

CONTENTS

	Page	
SUMMARY	1	1/A5
INTRODUCTION	1	1/A5
OVERVIEW	2	1/A6
Method of Analysis	2	1/A6
Blade geometric model	2	1/A6
Numerical model	3	1/A7
General Program Description	3	1/A7
DETAILED PROGRAM PROCEDURE	5	1/A9
Block Data NGASDAT	6	1/A10
Main Program NTTACT	6	1/A10
Subroutine FLOWS	7	1/A11
Subroutine FLSPLT	7	1/A11
Subroutine GASTBL	8	1/A12
Subroutine GAUSS	8	1/A12
Subroutine GETIN	8	1/A12
Function HCFRCD	9	1/A13
Subroutine HCOOL	9	1/A13
Subroutine HCPINS	9	1/A13
Subroutine INPRT	9	1/A13
Subroutine PARRAY	9	1/A13
Subroutine PLNUM	10	1/A14
Subroutine PLOTMF	10	1/A14
Subroutine PREP	10	1/A14
Subroutine TARRAY	10	1/A14
Subroutine TCOEF	11	1/B1
Subroutine THRCON	11	1/B1
Subroutine WROUT	11	1/B1
DICTIONARY	11	1/B1
PROGRAM LISTING	53	1/E1
REFERENCES	150	2/E1

Item 830-H-15 NAS 1.60:1391

MAR 7 1979

NASA Technical Paper 1391

COMPLETED
ORIGINAL

**TACT1, A Computer Program for
the Transient Thermal Analysis
of a Cooled Turbine Blade or Vane
Equipped With a Coolant Insert
II - Programmers Manual**

Raymond E. Gaugler

JANUARY 1979

NASA

165

NASA Technical Paper 1391

**TACT₁, A Computer Program for
the Transient Thermal Analysis
of a Cooled Turbine Blade or Vane
Equipped With a Coolant Insert
II - Programmers Manual**

Raymond E. Gaugler
Lewis Research Center
Cleveland, Ohio



National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

1979

Blank Page

SUMMARY

A FORTRAN IV computer program to calculate transient and steady-state temperatures, pressures, and flows in a cooled turbine blade or vane with an impingement insert has been developed and is described in this report. Coolant-side heat-transfer coefficients are calculated internally in the program, with the user specifying one of three modes of heat transfer at each station: (1) impingement, including the effect of crossflow; (2) forced-convection channel flow; or (3) forced convection over pin fins. Additionally, a limited capability to handle film cooling is available in the program. It is assumed that spent impingement air flows in a chordwise direction and is discharged through a split or drilled trailing edge and through film-cooling holes. The program does not allow for radial flow of the spent impingement air. The use of film cooling is restricted by a numerical model requirement for a continuous coolant-channel flow.

Input to the program includes a description of the blade geometry, coolant-supply conditions, outside thermal boundary conditions, and wheel speed. The user can divide the blade by chordwise cuts into as many as 15 slices and can divide each slice into as many as 79 stations around the blade. Each station in turn consists of four calculational nodes through the wall and one in the coolant channel. The blade wall can be two layers of different materials, such as a ceramic thermal-barrier coating over a metallic substrate. Program output includes the temperature at each node, the coolant pressures and flow rates, and the coolant-side heat-transfer coefficients.

INTRODUCTION

As core turbine-engine operating conditions become more severe, it becomes more difficult to effectively cool blades and vanes. Advanced transient thermal calculational techniques are needed to design reliable turbine blades. However, there appears to be no computer program generally available that uses these advanced techniques in combining the required heat-transfer and coolant-flow-distribution calculations. Thus, it was decided to create a computer program that would perform both transient and steady-state heat-transfer and coolant-flow analyses for a cooled blade, given the outside hot-gas boundary conditions, the coolant inlet pressure or flow rate, the geometry of the blade shell, and the cooling configuration.

The resulting program, TACT1, can handle a turbine blade or vane that is equipped with a central coolant-plenum insert from which coolant air flows through holes to impinge on the inner surface of the blade shell or directly into the trailing-edge region. It

is assumed that the spent impingement air then flows chordwise and is dumped through a split or drilled trailing edge and through film-cooling holes. The blade is modeled by dividing it by chordwise planes into as many as 15 slices, with each slice having as many as 79 calculational stations around the blade. Temperatures at each station are calculated for four points through the wall and one in the coolant channel. Included in this model is the capability to analyze a blade with a ceramic thermal-barrier coating. The ability of the program to model film cooling is limited by the numerical flow analysis requirement for a continuous coolant-channel flow.

The TACT1 program is used at the NASA Lewis Research Center on an IBM TSS/360-67 computer. The source program consists of approximately 6000 lines of code and the program requires about 60 000 words of storage. Typical running times for the program are 1.4 seconds of central processor unit (CPU) time per calculational station for a steady-state run and 0.4 second of CPU time per station per time step for a transient run.

The TACT1 program is reported in two parts. This report, part II, is a programmers manual and includes a complete program listing and a detailed description of the procedure. Part I (ref. 1) is a users manual and contains all the information necessary to run the program: a detailed description of the input, the method of solution, and the output as well as a sample problem.

OVERVIEW

Method of Analysis

The details of the analytical method are presented in part I (ref. 1). The blade model used in the analysis is described briefly in this section.

Blade geometric model. - The key to creating a usable computer program is to have as simple a geometric model as possible for the system being analyzed. In this program, the emphasis is on a blade or vane with a central coolant plenum and chordwise flow of the coolant after impingement. Therefore, it was decided that the primary calculational direction would also be chordwise. The blade is divided into layers that are bounded by chordwise cuts through the blade, as shown in figure 1. Each slice is treated separately in the program, with radial heat conduction in the wall the only communication between layers.

Figure 2 gives the details of the geometric model for a single blade slice and shows the breakdown of the blade or vane into calculational stations and nodes. Each calculational station consists of five nodes: one at the wall outer surface, one at the interface between the coating and blade metal, one at a point midway through the wall metal, one at the wall inner surface, and one in the middle of the coolant channel.

For input to the program, the following basic elements of the geometry are needed for each station: (1) the thicknesses of the wall coating and wall metal and the coolant-channel width, (2) the distance of each node from the adjacent lower-numbered node, and (3) the radial span for this slice. In addition, depending on the mode of heat transfer specified, the user must supply impingement-hole diameter and spacing or pin-fin diameter and spacing. Thermal properties of the blade materials must also be specified. The input is described in detail in reference 1.

Numerical model. - The numerical solution for the temperatures throughout the blade involves writing a transient energy-balance equation for each node and forming a set of equations to be solved for the temperature distribution. Similarly, the coolant pressure distribution is determined by writing the transient momentum equation for flow between adjacent fluid nodes and solving the resulting set of equations for static pressures.

The nodal energy balances are linearized, one-dimensional heat-conduction equations at the wall outer-surface node, at the coating-metal interface, and at the wall inner-surface node. At the midmetal node, a linearized, three-dimensional, heat-conduction equation is used. In the coolant channel, energy and momentum equations for one-dimensional compressible flow including friction and heat transfer are written for the elemental channel length between two coolant nodes. The equations used are presented in reference 1.

General Program Description

The TACT1 program is capable of performing a transient analysis as well as a steady-state analysis. In the case of a transient, the program first performs steady-state calculations to determine the initial conditions for the transient.

Figure 3 shows a schematic of the TACT1 calculational scheme. There are three basic, nested calculational loops that must converge for a steady-state solution to be reached. These loops are labeled A, B, and C in figure 3. The program begins a steady-state analysis with the coolant-supply pressure and total coolant flow fixed. The impingement flow is initially assumed to split uniformly at the leading-edge stagnation station, station 1. All coolant flows for the slice under consideration are calculated first, based on the current pressure distribution. The temperatures at each node are then calculated by solving simultaneously the energy equations presented in reference 1. The pressures at each coolant node are calculated by solving simultaneously the momentum equations presented in reference 1. This cycle, loop A in figure 3, is repeated until the pressure distribution no longer changes. The flow split between suction- and pressure-side coolant channels is then checked by comparing the pressures at the ends of the two channels. If they do not match, the impingement flow split at the leading edge

is adjusted and the inner loop calculations are repeated. This adjustment comprises loop B in figure 3. Once the proper flow split is achieved, the program moves up the blade to the next slice and repeats this sequence. After all N slices have converged, the total coolant mass flow used is compared with the inlet coolant mass flow. If there is an imbalance, either the inlet flow or the supply pressure is adjusted, depending on which was specified in the input; and the calculations start over. This is loop C in figure 3. Once the overall coolant mass flow balance is satisfied, the steady-state solution is complete and the transient calculations begin. During a transient calculation, loop B is bypassed because the coolant flow-split is primarily a function of blade geometry. Loop C is also bypassed because the inlet coolant mass flow rate at a given time is estimated based on the coolant mass flow used at the previous time step and the change in supply pressure.

The TACT1 subprograms and the calling relations are shown in figure 4. Block data subprogram NGASDAT contains air properties, for use in TACT1, tabulated as functions of temperature at a pressure of 20 atmospheres from reference 2. This subprogram must be loaded before execution of the program. The main program, NTTACT, calls other subroutines in their proper order.

The first call from NTTACT is to GETIN, a subroutine that controls the reading, storing, and printing of input data. Subroutine GETIN calls INPUT to print the input data if the user specifies INEDIT > 0. Subroutine INPRT has a call to PREP to put the input data in its proper form for use. All data are input by using a NAMELIST format.

After the input data have been read, the number of time steps, NTYM, to be used in the transient is determined in NTTACT. If only a steady-state solution is to be calculated, NTYM = 1. Time-dependent boundary conditions are then evaluated, with the initial entries assumed to be steady-state values. Then NTTACT loops through the blade, calling on subroutines PLNUM, PREP, and TCOEF for each slice. The first time through is a steady-state calculation.

Subroutine PLNUM calculates the pressure distribution in the impingement plenum for the current slice, given the inlet pressure and coolant flow-rate. PLNUM calls GASTBL for gas properties.

Subroutine PREP extracts the input data for the current slice from the input arrays.

Subroutine TCOEF controls loop A in figure 3, the iterative calculations of temperature and pressure for the nodes of the current slice. Each iteration in TCOEF requires calls to subroutines FLOWS, HCOOL, THRCOOL, TARRAY, PARRAY, and GAUSS.

Subroutine FLOWS computes the impingement jet flow rates, coolant-channel mass flow rates, and channel Mach numbers for each station around the blade, given the plenum pressure and temperature and the current pressure distribution in the coolant channel. FLOWS calls GASTBL for gas properties.

Subroutine HCOOL is called to calculate coolant heat-transfer coefficients for all the stations of this slice, based on the latest values of mass flow rate. HCOOL calls function HCFRCD to calculate forced-convection heat-transfer coefficients and GASTBL for gas properties.

Subroutine THRCON determines the wall thermal conductivity from the input table of conductivity as a function of temperature.

Subroutine TARRAY sets up the array of coefficients for the conduction and convection equations for each node. Calls are made to HCPINS for pin-fin heat-transfer coefficients, to HCFRCD for forced-convection heat-transfer coefficients, and to GASTBL for gas properties. TCOEF calls subroutine GAUSS to solve the set of equations for the temperature at each node.

Subroutine PARRAY sets up the array of coefficients for the momentum equations in the coolant channels and TCOEF calls subroutine GAUSS to solve the set of equations for the pressure at each coolant node.

After a new set of temperatures and pressures has been determined, convergence is checked by using the coolant-channel pressure at the blade leading edge. If this pressure stays within a tolerance band for four successive iterations, convergence is accepted. Once convergence is achieved, TCOEF calls subroutine FLSPLT to check the coolant flow-split between the pressure and suction sides. This is loop B in figure 3. Initially, the impingement jet flow at the forward stagnation station is assumed to split evenly between the suction- and pressure-side channels. If the coolant-channel pressures at the end of the impingement insert do not match, the flow split at the forward stagnation station is adjusted to increase the flow to the channel with the higher pressure at the end of the insert, and iteration loop A is repeated. Once a satisfactory flow split has been achieved, TCOEF calls subroutine WROUT to print the output for this slice and calls subroutine PLOTMF if there is to be graphical output. After NTTACT has calculated all blade slices, the total coolant mass flow is compared with the impingement-plenum inlet mass flow rate used to start the calculations. If the two flow rates are not close enough, the inlet mass flow or supply pressure is adjusted and the calculations are repeated. This is loop C in figure 3.

When the initial steady-state solution has been completed, the transient calculations are started. The transient is continued until the time reaches the specified maximum.

Subroutine PLOTMF makes use of a TSS/360 graphics package at the NASA Lewis Research Center to plot temperature and pressure distributions for the blade.

DETAILED PROGRAM PROCEDURE

Table I lists the names of each of the subprograms in TACT1, the corresponding

TSS/360 source module names, the COMMON blocks used in each, the names of the subroutines called by each, and the names of subroutines calling each. Table II is a cross-reference listing of named COMMON blocks and the subprograms using them. This section gives a detailed description of each subprogram used in TACT1. All variable names used are defined in the section DICTIONARY. The BLOCK DATA subprogram and the MAIN PROGRAM are discussed first and then each subprogram is described, in alphabetical order.

Block Data NGASDAT

A BLOCK DATA subprogram, NGASDAT, is used to provide a table of gas properties to the program. The properties are put in the array GS through a DATA statement with $5 \times NG$ entries, where NG is the number of table entries for each property. The first NG values are temperatures, the second are thermal conductivities, the third are specific heats, the fourth are Prandtl numbers, and the final NG values are viscosities. The property values included are taken from reference 2 at a pressure of 20 atmospheres.

Main Program NTTACT

The MAIN PROGRAM for TACT1, NTTACT, has overall control of the program. Figure 5 is a flow chart for NTTACT. During initialization, a call is made to a system subroutine, TIME, to get a unique label to be used to identify the plotted output for a given run. After the call to GETIN, where all the input data are read, NTTACT initiates the solution procedure by searching the transient boundary condition tables and using linear interpolation to extract the values for the current time. The next step is to begin the loop, labeled C in figure 3. The solution progresses from hub to tip. For each slice, NTTACT calls PLNUM to calculate coolant-supply conditions; PREP to extract the input data from the input tables; and TCOEF to calculate flows, temperatures, and pressures. After the return from TCOEF, NTTACT updates the total amount of coolant used, WUSED, by adding the amount used in the current slice, WIM. The amount of coolant-plenum flow available for the remaining slices, WPLEN, is updated by subtracting WIM. After all slices have been done, the overall amount of coolant used is printed and then checked against the assumed coolant flow-rate. If the absolute value of the difference, EXCESW, is more than 1 percent of the assumed flow, the assumed flow or the supply pressure is adjusted and the calculations are repeated. For transient runs, after the initial steady-state coolant-flow balance, there are no more iterations on coolant flow. Instead the flow for a given time step is based on the actual flow used in

the preceding time step and on the ratio of supply pressure for the two steps. Finally, once all loops have been completed, NTTACT calls PLOTMF to get a final summary plot of blade temperatures.

Subroutine FLOWS

Subroutine FLOWS is a routine to calculate the flow rates through all impingement and film-cooling holes, the friction factor in the coolant channels, and film-cooling effectiveness. FLOWS makes use of the current impingement-plenum mean pressure and temperature and coolant-channel pressure and temperature distributions. The impingement jet flow-rate, WJ, is calculated for each station in the forward region and checked against the choked flow-rate, WCR. If WJ is greater than WCR, then WJ is set equal to WCR. If there is any film cooling on the blade, the film-cooling flow rates in the forward region, WFC, are also calculated. Then, the coolant-channel flow rates, WCROS, are computed by considering a mass balance between stations, as illustrated in figure 6. Once the forward-region coolant flows have been determined, the Reynolds numbers - RE for the coolant channel, and REFC for the film-cooling flow - and the square of the coolant Mach number, AM2, are computed for each forward station.

The next step is to calculate the amount of coolant, WDUMP, dumped directly into the trailing-edge region from the coolant plenum. Then the total amount of coolant used for this slice, WIM, is determined by summing the impingement jet flows and WDUMP. Following this, the flows in the trailing-edge region are computed, with the coolant flow being reduced by the amount of any film-cooling flow. Then, trailing-edge-region values of RE, REFC, and AM2 are calculated.

After all the coolant flow-rates are determined, the friction factor, FF, is calculated at each station. Finally, if there is any film cooling used, the film effectiveness is calculated by using the method of reference 3.

Subroutine FLSPLT

Subroutine FLSPLT is used to determine the location of the stagnation impingement jet, station JS, and the fraction of that jet's flow that splits to each side of the blade, DELTAN. Figure 7 is a detailed flow chart for subroutine FLSPLT. The primary variable carried into FLSPLT is the pressure-match parameter, EPSN, which is defined as

$$EPSN = \frac{(P(2, ISLICE, NFWD-1) - P(2, ISLICE, NFWD))}{P(2, ISLICE, NFWD-1)} \quad (1)$$

where the pressures are as illustrated in figure 8.

The magnitude and sign of EPSN are used to determine the adjustment of the stagnation impingement-jet row location and the fraction of that jet that splits to the suction-side channel. Initially, the stagnation jet row is located at station 1 and the split is $\text{DELTAN} = 0.50$. If EPSN is positive, DELTAN is set to 0.75 to increase the flow down the suction-side channel; if EPSN is negative, DELTAN is set to 0.25 to increase the flow down the pressure-side channel. For subsequent entries into FLSPLT, the value of DELTAN is adjusted by passing a straight line through the last two points on a plot of EPSN versus DELTAN and picking the value of DELTAN where this line crosses the axis at $\text{EPSN} = 0$. If this intercept falls outside the DELTAN range of 0 to 1, the stagnation station, JS, must be moved to an adjacent station and DELTAN set to 0.50. Once a sign change is observed in EPSN, a fine-tuning process is triggered in FLSPLT. In this case, the values of DELTAN and EPSN for the iteration preceding the sign change are saved and used as one of the points of the straight-line interpolation scheme for all subsequent iterations.

Subroutine GASTBL

Subroutine GASTBL is used to interpolate in the array GS for gas properties, given the absolute temperature. Linear interpolation is used.

Subroutine GAUSS

Subroutine GAUSS is a routine that uses Gaussian elimination to solve a set of simultaneous equations. The array of coefficients, TCOF, is in the form of a compressed, augmented band matrix. That is, only the matrix elements within the band and the constants from the right side are stored in TCOF. The matrix band width, K, is determined by the node-numbering system used. In TACT1, the temperature calculations require a band width of 23 elements, and the pressure calculations require 19.

Subroutine GETIN

Subroutine GETIN is a routine used to initialize input-data default values and to read and store input data. Input is in NAMELIST form as described in reference 1. The entire data set is read, and the input variables for each slice are stored in two arrays: INDCHN for integer data, and CHANL for real-number data. If the input is provided in SI units, subroutine GETIN converts it to U.S. customary units for internal use. If the user specifies $\text{INEDIT} > 0$, GETIN calls subroutine INPRT to print out the input data.

Function HCFRCD

Function subprogram HCFRCD is a routine to calculate a turbulent, forced-convection heat-transfer coefficient for channel flow as described in reference 1.

Subroutine HCOOL

Subroutine HCOOL is a routine containing the correlations for impingement heat transfer. The first part of HCOOL deals with leading-edge-region impingement cooling. In this part, the inner surface length from the stagnation impingement jet to the end of the leading-edge impingement region is determined and then used in a correlation to compute the average heat-transfer coefficient in this region. Beyond this region, for stations starting at ICOR, calculations are done by using an impingement-with-crossflow correlation.

Subroutine HCPINS

Subroutine HCPINS is a routine to calculate coolant-side heat-transfer coefficients in regions of the blade equipped with pin fins. In addition, the effective heat-transfer area, which accounts for the pin surface area and the pin-fin effectiveness, is calculated.

Subroutine INPRT

Subroutine INPRT is a routine to print a listing of the input data. Also, INPRT sets up the initial temperature distribution in the blade. Subroutine PREP is called for each slice to extract input data from the arrays INDCHN and CHANL.

Subroutine PARRAY

Subroutine PARRAY is a routine to set up the matrix to be solved for coolant-channel pressure distribution. The equations used are detailed in reference 1.

The array of coefficients generated in PARRAY, TCOF, is in the form of a compressed, augmented band matrix. Coefficients that would be on the main diagonal of the full matrix are stored in column 10 of the TCOF array. The terms from the right side of the equations are stored in column 20 of TCOF.

Subroutine PLNUM

Subroutine PLNUM is a routine to calculate the pressure and temperature distributions in the central coolant plenum. The mean plenum static pressure and temperature for each slice are used as the supply conditions for the impingement jets. The total temperature and pressure at the outlet of one plenum slice are used as input for the next slice.

There are five arguments used in the call statement for PLNUM: WXX is the mass flow rate into this plenum slice; PXX and TXX are the calculated, average static pressure and temperature; and PTEXTIT and TTEXTIT are total temperature and pressure, respectively. Going into the subroutine, PTEXTIT and TTEXTIT are the values at the entrance to this slice. On return, they are the values at the exit of this slice.

Subroutine PLOTMF

Subroutine PLOTMF is a routine that plots TACT1 output. PLOTMF makes use of a TSS/360 graphics package at the NASA Lewis Research Center. For an installation without this specific package, this subroutine would have to be revised or bypassed.

PLOTMF plots temperature and pressure versus surface distance from station 1 for each slice of the blade for a steady-state case. For transients, a set of two summary plots is made for each time step: the plots contain temperatures for all slices on one graph.

Subroutine PREP

Subroutine PREP is a routine to extract input data from storage and put it in the form used in the calculations. In PREP, the hot-gas-side boundary condition tables are searched and linear interpolation is done to extract the boundary condition values at each calculation station at the given time.

Subroutine TARRAY

Subroutine TARRAY is a routine to set up the matrix to be solved for the temperatures in each slice. The equations used are detailed in reference 1.

The array of coefficients generated in TARRAY, TCOF, is in the form of a compressed, augmented band matrix. The 12th column of TCOF contains the elements that would be on the main diagonal of a full matrix. The terms from the right side of the equations are stored in column 24.

Subroutine TCOEF

Subroutine TCOEF is a routine that controls the calculations for flow rates, temperatures, and pressures. The first time TCOEF is entered for each slice an initial estimate of the coolant-channel pressure distribution is set up. TCOEF controls the iterations in loops A and B in figure 3. Loop A consists of calls to subroutines FLOWS, TARRAY, and PARRAY. The variable IVERGE is used to count the number of iterations in loop A. Convergence is checked by comparing the four most-recent values of coolant-channel pressure at the flow-split point. When the ratio of the maximum difference among these four to the difference between coolant-supply pressure and trailing-edge exit pressure is less than PCNVRG, loop A is complete. Then the flow split at the stagnation impingement jet, JS, is checked by subroutine FLSPLT and adjusted if necessary. Loop B involves repeating loop A for a new flow split. The variable IDELT is used to count the number of flow-split iterations in loop B. Once flow-split convergence is achieved, WROUT is called to print the output for the current slice.

Subroutine THRCON

Subroutine THRCON is a routine that takes the wall temperatures and searches for the thermal conductivity values in the input tables.

Subroutine WROUT

Subroutine WROUT is a routine to control the printing of the output from TACT1. Output units are the same as the input data units.

DICTIONARY

All the important variable names used in the TACT1 code are defined in this section. The only names not defined are locally used indices. All dimensioned variables include the dimensions. The dictionary also indicates the COMMON block or subroutine in which each variable is used.

Variable	Common	Subroutine	Definition
A(400)	TCO		cross-sectional area normal to chord-wise direction, in ² , accessed by node number
AA		GETIN	outer-surface length between stations, in., used for calculating interpolated values of TDLX(2), TDLX(3), and TDLX(5)
AA		PLNUM	coolant-plenum cross-sectional area, in ² , used in plenum pressure-drop calculations
AB		PLNUM	maximum Mach number in coolant plenum
AC(5)		GASTBL	array of interpolated values of gas properties
ACH		PLNUM	coolant-plenum choked-flow indicator
ADUMP	TCO		area of slot or jets dumping coolant directly into trailing-edge region, in ²
ADUMPC		INPRT	same as ADUMP, but converted to cm ² for input listing when input is in SI units
AHG		PREP	intermediate value of hot-gas-side heat-transfer coefficient, Btu/hr · ft ² · °F, used for interpolating in input table during a transient
AHTRN1		TARRAY	inner-surface area for heat-transfer purposes, in ²
AHTTR		HCPINS	total surface area in pin-fin channel, in ²
AINTRV		PLOTMF	floating-point form of number of temperature intervals in summary plots
AJ		PLNUM	floating-point form of indicator J - 1
AJET(80)	TCO		total area of impingement jets at each station, in ²

Variable	Common	Subroutine	Definition
AJET(80)		FLSPLT	total area of impingement jets at each station, in ² , carried into subroutine as argument
AKC(15, 80)	TCO		wall outer-coating thermal conductivity, Btu/hr · ft · °F
AKCTBL(20)	BOUND		input table of wall outer-coating thermal conductivity, Btu/hr · ft · °F, versus temperature, °F
AKW(15, 80)	TCO		wall metal thermal conductivity, Btu/hr · ft · °F
AKWTBL(20)	BOUND		input table of wall metal thermal conductivity, Btu/hr · ft · °F, versus temperature, °F
ALABL(7)		PLOTMF	array containing time and date label for identification of output plots
ALPH(12)		NTTACT	alphameric array used to uniquely identify output of each job
ALPHA	FRIC		constant used in friction factor calculations
ALPH2(4)		PLOTMF TCOEF NTTACT	time and date information, generated in NTTACT and passed to plotting subroutine as argument
AM		HCOOL	exponent on Reynolds number in Kercher-Tabakoff impingement correlation
AMC(20)		PLNUM	Mach number distribution in coolant plenum for a given slice
AMCHOK		FLWS	if any stations have a Mach number greater than 1.0, the value is saved in this variable and returned as an argument, to be printed by TCOEF
AMIN		FLWS HCPINS	area of coolant-flow channel at a given station, reduced by pin-fin blockage

Variable	Common	Subroutine	Definition
AM2(80)	TCO		array containing square of coolant-channel Mach number at each station, for a given slice
APG		PREP	intermediate value of hot-gas-side pressure, lbf/in^2 , used for interpolating in input table during transient
APLEN		GETIN	input value of coolant-plenum area for given slice, cm^2 (in^2)
APLN(15)	RADL		internal array to store plenum area for each slice, in^2
AP1		GASTBL	interpolating factor in gas property table lookup
AP2		GASTBL	$1.0 - \text{AP1}$
AQG		PREP	intermediate value of hot-gas-side heat flux, $\text{Btu/hr} \cdot \text{ft}^2$, used for interpolating in input table during transient
ASTG		TCOEF	inner-surface area under stagnation-point impingement jet, in^2
ATG		PREP	intermediate value of hot-gas-side temperature, $^{\circ}\text{R}$, used for interpolating in input table during transient
ATMAXP		PLOTMF	adjusted maximum temperature, $^{\circ}\text{F}$, used as high endpoint on output plots
ATMINP		PLOTMF	adjusted minimum temperature, $^{\circ}\text{F}$, used as low endpoint on output plots
ATYME		PLOTMF	value of time in transient, sec, used on output plots for identification
AVRGA		PARRAY	area ratio used in momentum equation at entrance to trailing edge
AZ		PLNUM	dummy variable, used as either diameter-area ratio or flow adjustment

Variable	Common	Subroutine	Definition
A1		FLows	interpolating factor in friction factor calculation in transitional Reynolds number range
A1		TARRAY	upstream half of inside-wall heat-transfer area, in^2 , associated with coolant-channel node
A2		FLows	interpolating factor in friction factor calculation in transitional Reynolds number range
A2		TARRAY	downstream half of inner surface heat-transfer area, in^2 , associated with coolant-channel node
A3		TARRAY	same as A1, but on opposite wall, only used in trailing-edge region
A4		TARRAY	same as A2, but on opposite wall, only used in trailing-edge region
B		GETIN	ratio of length to thickness, used along with AA to calculate interpolated values of TDLX(2), TDLX(3), and TDLX(5)
B(20)		PLNUM	spanwise static-temperature distribution, $^{\circ}\text{R}$, in coolant plenum for given slice
BC		GETIN	NAMelist name
BCHGP(1000)	BOUND		input table of hot-gas, pressure-side heat-transfer coefficients, $\text{W}/\text{m}^2 \cdot \text{K}$ ($\text{Btu}/\text{hr} \cdot \text{ft}^2 \cdot ^{\circ}\text{F}$)
BCHGS(1000)	BOUND		input table of hot-gas, suction-side heat-transfer coefficients, $\text{W}/\text{m}^2 \cdot \text{K}$ ($\text{Btu}/\text{hr} \cdot \text{ft}^2 \cdot ^{\circ}\text{F}$)
BCPGP(1000)	BOUND		input table of hot-gas, pressure-side relative static pressure, kPa (lbf/in^2)
BCPGS(1000)	BOUND		input table of hot-gas, suction-side relative static pressure, kPa (lbf/in^2)

Variable	Common	Subroutine	Definition
BCQGP(1000)	BOUND		input table of hot-gas, pressure-side heat flux, W/m^2 (Btu/hr · ft ²)
BCQGS(1000)	BOUND		input table of hot-gas, suction-side heat flux, W/m^2 (Btu/hr · ft ²)
BCTGP(1000)	BOUND		input table of hot-gas, pressure-side temperature, K (°F)
BCTGS(1000)	BOUND		input table of hot-gas, suction-side temperature, K (°F)
BCTIME(50)	BOUND		input table of time at which transient input tables are specified, sec
BCXP(400)	BOUND		input table of outer-surface, pressure-side locations at which hot-gas conditions are input, cm (in.)
BCXS(400)	BOUND		input table of outer-surface, suction-side locations at which hot-gas conditions are input, cm (in.)
BES		HCOOL	equivalent slot width, in., used in leading-edge impingement correlation
BETA	FRIC		constant used in friction factor calculations
BETA1		PLNUM	square of pressure at inlet to coolant plenum for given slice, $(lbf/in^2)^2$
BETTA(20)		PLNUM	spanwise static-pressure distribution in coolant plenum, lbf/in^2
BTA	TCO		indicates type of hot-gas boundary condition
C		FLOWS GASTBL HCFRCD HCOOL HCPINS PLNUM TARRAY	gas thermal conductivity, Btu/hr · ft · °F

Variable	Common	Subroutine	Definition
CD	TCO		impingement-jet discharge coefficient
CDEN(2)	UNITS		conversion factor for density units
CD1(200)		NTTACT	dummy variable used to print selected intermediate temperature values
CEXCSW		NTTACT	amount of excess coolant flow, in SI units, kg/hr
CGASC(2)	UNITS		conversion factor for gas constant
CH(15)		PLNUM	coolant-channel choking indicator
CHANL(8000)	SPECL		array for storing input data
CHANLS		GETIN	NAMELIST name
CHFLX(2)	UNITS		conversion factor for heat-flux units
CHTC(2)	UNITS		conversion factor for heat-transfer-coefficient units
CIMP1	IMPCOR		user-supplied constants for general impingement correlation
CIMP2			
CIMP3			
CIMP4			
CIMP5			
CIMP6			
CIMP7			
CINCH(2)	UNITS		conversion factor for length units
CMSFL(2)	UNITS		conversion factor for mass flow rate units
CNUM(80)	TCO		number of impingement jets at each station for given slice
CONDUCT		HCOOL	coolant-air thermal conductivity, Btu/hr · ft · °F
CONTRL		GETIN	NAMELIST name
CP	TCO		gas specific heat at constant pressure, Btu/lbm · °F

Variable	Common	Subroutine	Definition
CPC(80)	PRPS		coolant specific heat at constant pressure at each coolant node for given slice, Btu/lbm · °F evaluated at a mean temperature between bulk coolant temperature and wall temperature
CPIM		NTTACT PLOTMF	mean impingement-plenum pressure for given slice, in SI units, kPa
CPM		FLows	hot-gas-stream specific heat at constant pressure, Btu/lbm · °F
CPO	PRPS		specific heat at constant pressure, Btu/lbm · °F, evaluated at impingement-jet supply temperature
CPRSR(2)	UNITS		conversion factor for pressure units
CRHOVG(2)	UNITS		conversion factor for density × velocity units
CRITR		FLSPLT	coolant flow-split convergence criterion
CSPHT(2)	UNITS		conversion factor for specific-heat units
CTCON(2)	UNITS		conversion factor for thermal conductivity units
CTMPF(2)	UNITS		conversion factor for temperature units
CT0G		NTTACT	mean impingement-plenum static temperature for given slice, in SI units, K
CURV		TARRAY	factor to account for wall curvature in heat-conduction equations
CVISC(2)	UNITS		conversion factor for viscosity units
CWPLEN		NTTACT	coolant-plenum flow rate at entrance to given slice, in SI units, kg/hr
CWUSED		NTTACT	total amount of coolant air used, in SI units, kg/hr
CX		PLNUM	function of isentropic exponent k , $-(k + 1)/[2(k - 1)]$

Variable	Common	Subroutine	Definition
C1		PLNUM	function of isentropic exponent k , $2k/(k - 1)$
C3		PLNUM	computed constant involving wheel speed and isentropic exponent
C3		FLAWS	ratio of specific heats at constant pres- sure, coolant to hot gas
C5		PLNUM	computed constant involving isentropic exponent and gas constant
C6		PLNUM	function of isentropic exponent k , $(k - 1)/2$
C7		PLNUM	computed constant involving isentropic exponent and gas constant
C8		PLNUM	computed constant involving isentropic exponent and gas constant
D		PLNUM	convergence parameter in coolant- plenum pressure calculations
DD		PLNUM	coolant-plenum hydraulic diameter, in
DEH		HCOOL	hydraulic diameter of equivalent slot, in., used in leading-edge impingement correlation
DELAST		FLSPLT	variable used to save flow-split fraction at which pressure-match parameter, EPSN, changes sign
DELTA	FRIC		constant used in friction factor calcula- tion
DELTAN(15)		FLAWS FLSPLT HCPINS TARRAY TCOEF WROUT	fraction of stagnation-point impingement- jet flow that splits to suction-side coolant-flow channel for each slice
DELTAO		FLSPLT	value of DELTAN from previous flow- split iteration

Variable	Common	Subroutine	Definition
DENOM		NTTACT	intermediate variable used in time interpolation of some boundary conditions
DH(80)	TCO		coolant-channel hydraulic diameter at each station, in.
DHF(80)	TCO		effective diameter of film-cooling hole at each station, in., defined as hydraulic diameter of one hole multiplied by square root of number of holes at station
DHJ(80)	TCO		actual hydraulic diameter of an impingement hole at each station, in.
DHYD		GETIN	input value of coolant-plenum hydraulic diameter for a slice, cm (in.)
DIFN		TCOEF	pressure difference parameter used in checking convergence
DIFO		TCOEF	maximum pressure difference parameter used in checking convergence
DIFTOL		PLNUM	tolerance on coolant-plenum pressure-drop calculations
DIMP1 DIMP2 DIMP3 DIMP4 DIMP5 DIMP6	IMPCOR		user-supplied constants for leading-edge impingement correlation
DLTAOP		FLSPLT	best value of DELTAN in the event of an unstable flow split
DLTYME	TRNSNT		time step used in transient calculations, sec
DLX(400)	TCO		chordwise distance from each node to adjacent upstream node, in.
DP(80)	PRPS		diameter of pin fins at each station, in.

Variable	Common	Subroutine	Definition
DPLN(15)	RADL		coolant-plenum hydraulic diameter for each slice, in.
DR		PLNUM	radial length increment in coolant-plenum calculations, in.
DR2		PLNUM	radial increment squared, in ²
DUMR1(80)	PRPS		dummy variable, not currently used
DUMR2(80)	PRPS		dummy variable, used to carry impingement-jet Reynolds number to output subroutine
DUMTER		PARRAY	intermediate variable in momentum equation involving coolant dumped directly into trailing edge
DUM1(10)		INPRT WROUT	dummy variables used to print input listings and program output
DUM2(10)		INPRT WROUT	
DUM3(10)		INPRT	
DUM4(10)			
DUM5(10)			
DUM6(10)			
DUM7(10)			
DUM8(10)			
DUM9(10)			
DUM10(10)			
DUM11(10)			
DUM12(10)			
DUM13(10)			
DUM14(10)			
DUM15(10)			
DUM16(10)			

Variable	Common	Subroutine	Definition
DUM17(10)		INPRT	dummy variables used to print input listings and program output
DUM18(10)		↓	↓
DUM19(10)			
DUM20(10)			
DUM25(10)			
DUM52(10)			
DUM53(10)			
DUM55(10)			
DX		PLNUM	spanwise step size used in calculating coolant-plenum pressure and temperature distributions
DXTEMP		PLNUM	variable to temporarily hold DX
DX1		TARRAY	path length between midwall node and adjacent upstream midwall node, in.
DX10		TARRAY	path length between outer coating - wall junction node and adjacent downstream outer coating - wall junction node, in.
DX2		TARRAY	path length between midwall node and adjacent downstream midwall node, in.
DX3		TARRAY	path length between outer-surface node and adjacent upstream outer-surface node, in.
DX4		TARRAY	path length between outer-surface node and adjacent downstream outer-surface node, in.
DX5		TARRAY	path length between inner-surface node and adjacent upstream inner-surface node, in.
DX6		TARRAY	path length between inner-surface node and adjacent downstream inner-surface node, in.

Variable	Common	Subroutine	Definition
DX7		TARRAY	path length between coolant node and adjacent upstream coolant node, in.
DX9		TARRAY	path length between outer coating - wall junction node and adjacent upstream outer coating - wall junction node, in.
D1		PLNUM	computed constant used in coolant-plenum pressure equations to account for effect of pumping due to wheel rotation
D2		PLNUM	computed constant used in coolant-plenum temperature equations to account for effect of pumping due to wheel rotation
E		PLNUM	factor used in adjusting convergence rate in coolant-plenum calculations
EFAREA(80)		HCPINS TARRAY	effective area, in ² , for heat transfer at stations with pin fins, including pin-fin effectiveness for heat transfer
EFTVNS		HCPINS	pin-fin effectiveness
EMES(80)	FLMCOL		for film cooling, ratio of coolant mass flux to free-stream mass flux, multiplied by equivalent slot width
EML		HCPINS	term used in pin-fin effectiveness calculation
ENDEFF		TARRAY	term in heat-transfer equations to account for convection to rear edge of blade when heat-transfer coefficients are input
ENDFLX		TARRAY	term in heat-transfer equations to account for convection to rear edge of blade when heat flux is input
EPLAST		FLSPLT	variable used to save latest value of pressure-match parameter, EPSN

Variable	Common	Subroutine	Definition
EPS	FRIC		constant used in friction factor calculations
EPSMIN		FLSPLT	minimum value attained by pressure-match parameter, EPSN, for unstable flow split
EPSN		FLSPLT TCOEF	pressure-match parameter, defined as difference between suction- and pressure-side coolant-channel static pressures at end of insert, divided by suction-side coolant-channel static pressure
EPSO		FLSPLT	old value of pressure-match parameter, EPSN
ETAPRM		FLWS	film-cooling effectiveness based on ratio of enthalpy differences
EXCESW		NTTACT	amount of excess coolant flow, difference between inlet flow and that actually used, lbm/hr
FACTOR		TARRAY	special variable to adjust amount of energy carried in by an impingement jet for case of a calculation station adjacent to flow-split station
FF(80)	TCO		value of friction factor at each flow station
FILM		PARRAY	term to account for momentum carried off by film-cooling air
FILMW		TARRAY	total film-cooling flow from given coolant node
FLMEFF(80)	FLMCOL		film-cooling effectiveness based on ratio of temperature differences
FM		GAUSS	multiplying factor used in Gauss elimination scheme

Variable	Common	Subroutine	Definition
FUNP		PLNUM	statement function to calculate pressure difference in coolant plenum
FUNT		PLNUM	statement function to calculate temperature difference in coolant plenum
F1(20)		PLNUM	friction factor in coolant plenum
GAM	TCO		ratio of specific heats
GAMC(80)	PRPS		ratio of specific heats at each coolant-channel node
GAMO	PRPS		ratio of specific heats at coolant-supply conditions
GEO		GETIN	NAMelist name
GG		HCOOL	mass flux ratio, coolant crossflow to impingement-jet flow
GI		HCOOL	momentum flux ratio, coolant crossflow to impingement-jet flow
GMASs		HCOOL	mass flux from row of leading-edge impingement holes
GS(200)	GAAS		table of gas properties
G1		PLNUM	computed constant in coolant-plenum calculations, involving flow rate, gas constant, and specific heat at constant pressure
G2		PLNUM	computed constant in coolant-plenum calculations, involving flow rate and gas constant
HBAR		WROUT	average coolant-side heat-transfer coefficient for given slice, $W/m^2 \cdot K$ ($Btu/hr \cdot ft^2 \cdot ^\circ F$)
HC(80)	TCO		coolant-side heat-transfer coefficients at each station for given slice, $Btu/hr \cdot ft^2 \cdot ^\circ F$

Variable	Common	Subroutine	Definition
HCAL (4)		INPRT	alphanumeric array containing labels identifying type of coolant-side heat transfer
HG (80)	TCO		hot-gas-side heat-transfer coefficient at each station for given slice, $\text{Btu/hr} \cdot \text{ft}^2 \cdot ^\circ\text{F}$
HSTGMX		TCOEF	maximum physically possible value of coolant heat-transfer coefficient under stagnation jet, $\text{Btu/hr} \cdot \text{ft}^2 \cdot ^\circ\text{F}$
HUB1		TARRAY	term in conduction equation to account for specified hub temperature
HUB3		TARRAY	term in conduction equation to account for specified hub heat flux
HX		TARRAY	multiplying factor on coolant heat-transfer coefficient, initialized to 1.0 but may be changed dynamically
HYCOS		HCPINS	hyperbolic cosine term
HYSIN		HCPINS	hyperbolic sine term
IADJIN	SPECL		input variable that indicates which coolant-plenum supply property is to be held fixed
ICLK		PARRAY	indicates which side of blade a given station is on: 0 if suction side, 1 if pressure side
ICHNL		INPRT PREP	slice number, carried through as argument
ICHOKE		FLows PARRAY TCOEF	number of station that shows choked coolant flow
ICOMP		TARRAY	number of station adjacent to impingement flow-split station in pressure-side direction

Variable	Common	Subroutine	Definition
ICOMS	TCO	TARRAY	number of station adjacent to impingement flow-split station in suction-side direction
ICONV		FLSPLT TCOEF	indicator for convergence of flow-split iterations
ICOR			station at which use of impingement-with-crossflow correlation is to begin
IDELT		FLSPLT TCOEF WROUT	counter of number of flow-split iterations performed
IDN		PARRAY	downstream node number for coolant-channel pressure calculations
IDNS		PARRAY	downstream station number for coolant-channel pressure calculations
IDX		PARRAY	upstream node number for coolant-channel pressure calculations
IEND		GETIN	last point in CHANL array occupied by data for given slice
IFCP		FLAWS	indicator used in locating pressure-side film-cooling holes
IFCS		FLAWS	indicator used in locating suction-side film-cooling holes
IFILM	TCO		input indicator for film cooling
IFLU		PREP	coolant-channel node number
IFNL		PARRAY	number of coolant-channel nodes
IFNL		TCOEF	total number of stations, minus 1
IFSPLT		FLAWS	indicates in which direction film-cooling air flows from stagnation station
IGG(80)		HCOOL	array containing node numbers at which ratio of coolant crossflow to impingement-jet flow is out of Kercher-Tabakoff correlation range

Variable	Common	Subroutine	Definition
IGGC		HCOOL	counts number of entries in IGG array
IHC(80)	TCO		indicates type of coolant-side heat transfer at each station for given slice
IHCT		GETIN	input value of IHC for given station
IHUB	TCO		indicates type of boundary to be used at hub end of blade
II		HCOOL	coolant-channel node number
IIHCTZ		GETIN	locates IHC array in overall array INDCHN
ILEAD		HCOOL	last station in range of leading-edge impingement correlation
IMS		FLows	location of film-cooling hole preceding current station
INDCHN(2000)	SPECL		array for storing integer input data
INEDIT		GETIN INPRT	input variable to control listing of input data
INN		PREP	location of IHC data in INDCHN array
INSTRT		GETIN	starting point for storage of integer data in INDCHN array for given slice
INUM		NTTACT	number of stations on each side of blade
IN1		PREP	location of end of group of single-valued variables in INDCHN array
IPLOT	SPECL		input indicator to control plotting options
IPRES		WROUT	pressure-side, outer-surface node number
IPRSMN		WROUT	location of minimum outer-surface temperature on pressure side
IPRSMX		WROUT	location of maximum outer-surface temperature on pressure side

Variable	Common	Subroutine	Definition
IRE(80)		HCOOL	list of station numbers at which impingement-jet Reynolds number is out of Kercher-Tabakoff correlation range
IRL		PARRAY	coolant node number
ISBLOK	TCO		starting point in CHANL array of data for given slice
ISEN		TARRAY	indicates which side of blade station IS is on
ISENS		FLows	indicates which side of blade a given station is on
ISENS		TARRAY	indicates which side of blade a given trailing-edge region station is on
ISLICE	TCO		current slice number
ISTA		GETIN	input station number
ISTAT		WROUT	outer-surface node number for given station
ISTATD		WROUT	outer-surface node number immediately downstream of ISTAT
ISTB		GETIN	input station number
ISTRT		HCOOL	station at which use of Kercher- Tabakoff correlation begins
ISTRT		TCOEF	first station in trailing-edge region
ISUCMN		WROUT	location of minimum outer-surface tem- perature on suction side
ISUCMX		WROUT	location of maximum outer-surface tem- perature on suction side
ISUCT		WROUT	suction-side station number
ISUP		TARRAY	adjacent station, upstream of current station, IS
ISYM(5)		PLOTMF	data array containing plotting symbol codes

Variable	Common	Subroutine	Definition
ITDHFZ		GETIN	locates film-cooling-hole data in CHANL array for given slice
ITDHJZ		GETIN	locates impingement-hole data in CHANL array for given slice
ITDLXZ		GETIN	locates node spacing data in CHANL array for given slice
ITDPZ		GETIN	locates pin-fin diameter data in CHANL array for given slice
ITHKZ		GETIN	locates wall and channel thickness data in CHANL array for given slice
ITIP	TCO		indicates type of boundary to be used at tip end of blade
ITRBG		INPRT	first station in trailing-edge region
ITRBG		WROUT	first station in trailing-edge region
ITREO		WROUT	last outside node at trailing edge
ITRRZ		GETIN	locates radial position data in CHANL array for given slice
ITSPZ		GETIN	locates pin-fin spacing data in CHANL array for given slice
ITXNZ		GETIN	locates impingement-hole spacing data in CHANL array for given slice
IUNITS	UNITS		indicates system of units used for input data
IUNSTB		FLSPLT	indicates whether flow split is stable or not
IUP		PARRAY	upstream node number for coolant-channel pressure calculations
IUPS		PARRAY	upstream station number for coolant-channel pressure calculations
IVARS(12)		PLOTMF	array of integer plotting controls
IVERGE		TCOEF WROUT	pressure iteration loop counter

Variable	Common	Subroutine	Definition
IWR		GAUSS	control on debugging output of coefficient matrix
IWRITE	SPECL		input control on amount of printed output
IXAX		PLOTMF	logical variable with value .TRUE.
IYAX		PLOTMF	logical variable with value .FALSE.
I1		PREP	starting point in INDCHN array for integer data for given slice, also used as starting point for nodal data in CHANL array for given slice
I3		PREP	starting point in CHANL array for station data
JDIS		FLows	number of stations that impingement flow-split station is displaced from station 1
JHCAL		INPRT	indicates type of heat transfer at given station
JLSTM		THRCON	size of wall thermal conductivity tables
JNUMS		FLSPLT	indicates whether previous call to FLSPLT resulted in unstable flow split
JOUTRG		FLSPLT	indicates attempt to split more than 100 percent of flow to one side
JPIV		GAUSS	pivotal column in matrix to be reduced
JS		FLows FLSPLT HCOOL PARRAY TARRAY TCOEF WROUT	station number at impingement flow-split point

Variable	Common	Subroutine	Definition
JSENS		FLOWS GETIN FLSPLT PARRAY TARRAY TCOEF	indicates which side of blade impingement flow-split is on
JSGNCH		FLSPLT	indicates whether a sign change has occurred in EPSN
JSO(15)		TCOEF	array for saving converged-flow-split station number for each slice
JSOLDS(25)		FLSPLT	array to keep track of stations checked as flow-split stations
JTIMES		FLSPLT	indicates whether DELTAN is to be rough adjusted or fine tuned
K		GAUSS	total bandwidth of matrix to be solved
K		NTTACT	counter of number of overall coolant-flow iterations
KSIG		PLNUM	coolant-plenum iteration counter
L		TARRAY THRCON	midwall node number at given station
LCOOL		FLOWS HCFRCD HCOOL HCPINS TARRAY TCOEF WROUT	coolant-channel node number at given station
LCOOLP		HCPINS TARRAY	inner-surface node across coolant-channel from given station
LCUP		HCPINS TARRAY	coolant-channel node number upstream of given station
LCUPP		TARRAY	inner-surface node across coolant channel and upstream of given station

Variable	Common	Subroutine	Definition
LCUPS		TARRAY	inner-surface node upstream of given station
LDN		TARRAY	midwall node number downstream of given station
LN		FLWS HCFRCD HCOOL HCPINS TARRAY	inner-surface node number at given station
LJ		INPRT TARRAY THRCON	node number at junction of outer coating and wall metal at given station
LOUT		TARRAY THRCON	wall outer-surface node number at given station
LSP		PREP	starting point in BCXP array of data for given slice
LSS		PREP	starting point in BCXS array of data for given slice
LUP		TARRAY	midwall node number upstream of given station
MACH1		PLNUM	indicator to keep track of step-size change
MD1	SPECL		indicator to control special condensed, on-line output
MD2	SPECL		indicator that job is complete and summary plots are to be produced
MD3	SPECL		plot counter
MNBC		INPRT	maximum of NBCS and NBCP
N		GAUSS	number of rows in matrix to be solved
NAG		PLNUM	indicator whether calculations have progressed beyond initial station

Variable	Common	Subroutine	Definition
NBCP	BOUND		input number of boundary condition points on pressure side
NBCS	BOUND		input number of boundary condition points on suction side
NBFRP		INPRT	number of pressure-side boundary condition points preceding data for given slice at given time
NBFRS		INPRT	number of suction-side boundary condition points preceding data for given slice at given time
NBLKSZ	TCO		size of data block in CHANL array for given slice
NCC		PLNUM	loop counter
NCHAR		NTTACT	number of characters in ALPH2 array
NCOOL		PLOTMF	coolant node number
NEND		HCOOL	end of region that uses leading-edge impingement correlation
NFC		FLWS	number of station containing film-cooling holes
NFCSUP(80)	FLMCOL		array identifying node supplying film cooling to each downstream node
NFLUID(200)		INPRT	array of coolant-channel node numbers for each station
NFWD	TCO		number of stations in forward region
NG	GAAS		number of temperature entries in gas property table, GS
NGEO		GETIN	number of NAMELISTs /GEO/ to be read in
NINTRV		PLOTMF	number of temperature intervals in summary plots
NIT		TCOEF	counter of number of overall coolant-flow iterations

Variable	Common	Subroutine	Definition
NL		INPRT	output line counter
NLBLS		PLOTMF	number of points at which symbols are to be plotted
NMM		PLOTMF	midwall node number
NMW		NTTACT	outer-surface node number
NODM		WROUT	midwall node
NODOUT		GETIN	outer-surface node
NODSF		FLows GETIN INPRT PREP FLSPLT PARRAY TARRAY TCOEF	number of nodes in forward region
NODST		GETIN INPRT PREP PARRAY TARRAY TCOEF NTTACT	total number of nodes for given slice
NODSTM		INPRT	total number of nodes minus 4
NOS		FLows INPRT WROUT PLOTMF	outer-surface node number
NPRCP		INPRT	number of points in each pressure-side boundary condition array for times preceding current time
NPRCS		INPRT	number of points in each suction-side boundary condition array for times preceding current time

Variable	Common	Subroutine	Definition
NPRTP		INPRT	number of points in each pressure-side boundary condition array per time step
NPRTS		INPRT	number of points in each suction-side boundary condition array per time step
NPTS		PLOTMF	number of points on given plot
NROW		GAUSS	number of matrix row to be displayed by debug output
NSAVE		TCOEF	coolant node number just upstream of exit, location of TSAVE
NSLICE	TCO		current slice number
NSTA	TCO		number of stations per slice
NSTAPS		PLOTMF	number of stations on each side of blade
NSTNS		PLNUM	number of spanwise stations per slice in coolant plenum
NTBC		GETIN	number of entries in input BCTIME array
NTIMES		INPRT	number of entries in BCTIME array
NTTG		PREP NTTACT	time step number
NTYM		NTTACT	number of time steps in transient
NUMS		FLSPLT	counter to force at least four attempts at a good flow split
P(2, 15, 80)	TCO		pressure at each node, for two consecutive time steps, lbf/in ²
PAVG		FLOWS	coolant-channel static pressure, lbf/in ² , used in calculating impingement hole flow rates
PBAR		FLOWS	pressure used in calculating square of coolant-channel Mach number

Variable	Common	Subroutine	Definition
PCNVRG		TCOEF	pressure-difference convergence criterion
PD		FLAWS GASTBL HCFRCD HCOOL HCPINS PLNUM TARRAY	Prandtl number
PDTOG		HCOOL	Prandtl number based on coolant-supply temperature
PEX(400)	BOUND		input array containing tables of static pressure at trailing-edge coolant exhaust, lbf/in^2
PEXC		INPRT	static pressure at trailing-edge coolant exhaust in SI units, kPa, for given slice
PEXIT(15)	TCO		static pressures at trailing-edge coolant exhaust for each slice at given time, lbf/in^2
PEXOLD(15)		TCOEF	saved value of exhaust static pressure, lbf/in^2 , used in setting initial guess of pressure distribution for subsequent time step
PEXTT		PLNUM	total pressure at exit of coolant plenum for given slice, lbf/in^2
PG(80)	FLMCOL		array containing hot-gas-side static pressure, lbf/in^2 , at each station
PI		HCOOL	constant, 3.14159
PIM	TCO		impingement-supply pressure, lbf/in^2
PIMOLD(15)		TCOEF	saved value of impingement-supply pressure, lbf/in^2 , for each slice

Variable	Common	Subroutine	Definition
PIN(15)	RADL		coolant total pressure, lbf/in^2 , at entrance to each slice
PINS		HCPINS TARRAY	number of pin fins at given station
PIVOT		GAUSS	main diagonal term of row of matrix being solved
PLEGN(5)		PLOTMF	alphanumeric array to label pressure-data plots
PLTYME(2)		PLOTMF	alphanumeric variable to print transient time on each plot
POLD(15, 80)		TCOEF	saved values of coolant-channel pressure, lbf/in^2 , from previous iteration
PP		PLNUM	intermediate pressure term
PPLEN		NTTACT	impingement-supply pressure, lbf/in^2
PROD		HCOOL	intermediate calculation result
PROPS		GETIN	NAMELIST name
PSAV(5)		TCOEF	array to save last four values of pressure at flow-split station, used to check convergence
PTEMP		PLNUM	intermediate pressure term
PTEXTIT		PLNUM	coolant-plenum total pressure, lbf/in^2 : entrance value going into subroutine, exit value coming out
PTIN		NTTACT	coolant-supply pressure for a given time, lbf/in^2
PTIO(50)	BOUND		input array of coolant-supply pressure, kPa (lbf/in^2), as function of time, sec
PTIOC		INPRT	initial coolant-supply pressure in SI units, kPa
PTNOLD		NTTACT	previous value of PTIN, lbf/in^2
PT1		PLNUM	calculated coolant-plenum inlet total pressure, lbf/in^2

Variable	Common	Subroutine	Definition
PUMP(80)	TCO		term to account for coolant pumping due to wheel rotation
PUMTRM		PARRAY	term to account for coolant pumping due to wheel rotation
PXX		PLNUM	average static pressure, lbf/in ² , in coolant plenum for given slice
QG(80)	TCO		hot-gas heat flux to blade at each station, Btu/hr · ft ²
QHUB(80)	BOUND		heat flux conducted to blade wall from hub platform at each station, Btu/hr · ft ²
QHUBIN(400)	BOUND		input table of hub heat flux at each station as function of time, W/m ² (Btu/hr · ft ²)
QSNK(80)	TCO		term to account for heat removal from wall by film-cooling flow through wall
QTIP(80)	BOUND		heat flux from blade wall at tip for each station, Btu/hr · ft ²
QTIPIN(400)	BOUND		input table of tip heat flux at each station as function of time, W/m ² (Btu/hr · ft ²)
R	TCO		gas constant; value for air is build in, 53.35 ft-lbf/lbm · °R
RATIO		THRCON	interpolating fraction
RCHRD		TARRAY	dimensionless ratio of time increment to chordwise length increment squared at each station
RCHRDM		TARRAY	maximum value of RCHRD for a given slice
RCVRY		TARRAY	recovery factor
RE(80)	PRPS		coolant-channel Reynolds number at each station

Variable	Common	Subroutine	Definition
REFC(80)	FLMCOL		film-cooling flow Reynolds number at each station
REJ(80)		HCOOL	impingement-jet Reynolds number at each station
REJOVR(80)		HCOOL	array to save values of impingement-jet Reynolds number that are out of range of correlation
REY		PLNUM	coolant-plenum Reynolds number based on hydraulic diameter
RHOBAR		TARRAY	mean density in coolant channel, lbm/in^3
RHOC	TRNSNT		input density of outer coating, kg/m^3 (lbm/ft^3)
RHOM	TRNSNT		input density of wall metal, kg/m^3 (lbm/ft^3)
RHOVG(400)	BOUND		input table of hot-gas-side, free-stream mass velocity at each station as function of time, $\text{kg/m}^2 \cdot \text{sec}$ ($\text{lbm/ft}^2 \cdot \text{sec}$)
RHOVGA(80)	FLMCOL		hot-gas-side, free-stream mass velocity at each station for given slice, $\text{lbm/ft}^2 \cdot \text{sec}$
RI		GETIN	input value of radial location of coolant-plenum inlet for given slice, cm (in.)
RIN(15)	RADL		table of RI values for each slice, in.
RO		GETIN	input value of radial location of coolant-plenum exit for given slice, cm (in.)
ROINVC		HCOOL	intermediate term in impingement correlation, ft^3/lbm
ROINVJ		HCOOL	intermediate term in impingement correlation, ft^3/lbm
ROOT		PARRAY	intermediate term in pressure calculations

Variable	Common	Subroutine	Definition
ROUT(15)	RADL		table of RO values for each slice, in.
RR(80)	TCO		mean radial location of each station for a given slice
RRP		PLNUM	radial location, in.
RTEMP		PLNUM	radial location, in.
RTNARR(2)		PLOTMF	array containing maximum and minimum values of plot variables
RTRNV		TARRAY	dimensionless ratio of time increment to through-the-wall length increment squared
RTRNVM		TARRAY	maximum value of RTRNV for given slice
S(15)	TCO		span of each slice, in.
SEGMTS		PLNUM	number of segments in coolant plenum for given slice
SIGB		PLNUM	dummy variable used in coolant-plenum calculations
SIGC		PLNUM	dummy variable used in coolant-plenum calculations
SIGMA(20)		PLNUM	coolant velocity distribution in coolant plenum
SLEGN(5)		PLOTMF	alphameric array to label suction-side plots
SLP		HCPINS	mean pin-fin length at given station, in.
SP(80)	PRPS		pin-fin spacing at each station, in.
SPAN	TCO		radial span of given slice, in.
SPANC		INPRT	radial span of given slice in SI units, cm
SPHTC	TRNSNT		input specific heat of outer coating, J/kg · K (Btu/lbm · °F)
SPHTM	TRNSNT		input specific heat of wall metal, J/kg · K (Btu/lbm · °F)

Variable	Common	Subroutine	Definition
ST		HCOOL	Stanton number calculated from user-supplied impingement correlation
STANMX		HCOOL	Stanton number calculated from leading-edge impingement correlation
SV(3)		PLNUM	array to save values of SIGC
SYMBL(10)		PLOTMF	array of integers to be used as plot symbols
SYMBOL		PLOTMF	particular entry from SYMBL array
T(2, 15, 400)	TCO		calculated temperature at each node for each slice for two time steps, °F
TABOVE		TARRAY	midwall temperature at given station in slice above current slice, °F
TAU(400)	TCO		array of thickness values, in.
TBAR		FLows	coolant temperature, °R, used to calculate Mach number
TBAR		HCPINS	ratio of temperature drops in pin fins, pressure-side wall temperature minus mid-coolant-channel temperature to suction-side wall temperature minus mid-coolant-channel temperature
TBAR		WROUT	mean outer-surface temperature for given slice, °F
TBARMD		WROUT	mean midwall temperature, °F
TBELOW		TARRAY	midwall temperature at given station in slice below current slice, °F
TBULK		WROUT	overall blade bulk-metal temperature, °F
TC		THRCON	mean temperature of blade cladding material, °F
TCIN		NTTACT	coolant temperature, °F
TCOF(400, 30)	MATRX		array of coefficients to be solved for temperature or pressure

Variable	Common	Subroutine	Definition
TDHF		GETIN	function of film-cooling-hole size and spacing
TDHJ		GETIN	hydraulic diameter of impingement hole
TDLX(5)		GETIN	array containing lengths from nodes at upstream station to corresponding nodes at current station
TDP		GETIN	input pin-fin diameter, cm (in.)
TEM		INPRT	coolant-inlet absolute temperature, K ($^{\circ}$ R)
TEPS	TRNSNT		factor used to define time mean properties
TERM		FLSPLT	adjustment to flow-split parameter
TG(80)	TCO		hot-gas reference temperature at each station for given slice, $^{\circ}$ R
THETA1		TARRAY	intermediate terms in nodal energy equations
THETA2			
THETA3			
THETA4			
THETA5			
THETA6			
THETA8			
THETA9			
THK(3)		GETIN	input array of thickness values, cm (in.)
THUB(80)	BOUND		specified temperature at each blade hub station for given time, $^{\circ}$ F
THUBIN(400)	BOUND		input table of hub temperatures at each station as function of time, K ($^{\circ}$ F)
TIKLE(30)		GETIN	blank alphameric array used to initialize title array

Variable	Common	Subroutine	Definition
TIN(15)	RADL		inlet total temperature in coolant plenum for each slice, °F
TIP1		TARRAY	term in conduction equation to account for specified tip temperature
TIP3		TARRAY	term in conduction equation to account for specified tip heat flux
TITL		GETIN	NAMELIST name
TITLE(30)	SPECL		alphameric array of title information from input
TLABL1(21) TLABL2(9)		PLOTMF	alphameric arrays used to put input title on plots
TMAXP		PLOTMF	maximum pressure-side temperature to be plotted
TMAXS		PLOTMF	maximum suction-side temperature to be plotted
TMFRAC		PREP	time-interpolating function
TMINP		PLOTMF	minimum pressure-side temperature to be plotted
TMINS		PLOTMF	minimum suction-side temperature to be plotted
TMP		GASTBL	temperature used to determine gas properties, °R
TMP1		GASTBL	temperature used to determine gas properties, °F
TOTSPN		WROUT	total span of blade, in.
TP		PLNUM	intermediate temperature variable
TPLEN		NTTACT	mean impingement-plenum static temperature, °F
TPM(500)		PLOTMF	table of pressure-side, midwall temperatures for summary plots, K (°F)
TPMAX		WROUT	maximum pressure-side, outer-surface temperature for given slice, K (°F)

Variable	Common	Subroutine	Definition
TPMIN		WROUT	minimum pressure-side, outer-surface temperature for given slice, K ($^{\circ}$ F)
TPO(500)		PLOTMF	table of pressure-side, outer-surface temperatures for summary plots, K ($^{\circ}$ F)
TREDGE		TARRAY	intermediate term in nodal energy equation in trailing-edge region
TREPS		PARRAY TARRAY	same as TEPS
TRR		GETIN	input mean radial location of given station, cm (in.)
TRTRM		PARRAY	intermediate transient term in pressure equations
TRTRMC		TARRAY	intermediate transient term involving outer coating
TRTRMG		TARRAY	intermediate transient term involving coolant
TRTRMJ		TARRAY	intermediate transient term for coolant channel at entrance to trailing edge
TSAVE		TCOEF	temperature in coolant channel, just upstream of exit
TSM(500)		PLOTMF	table of suction-side, midwall temperature for summary plots, K ($^{\circ}$ F)
TSMAX		WROUT	maximum suction-side, outer-surface temperature for given slice, K ($^{\circ}$ F)
TSMIN		WROUT	minimum suction-side, outer-surface temperature for given slice, K ($^{\circ}$ F)
TSO(500)		PLOTMF	table of suction-side, outer-surface temperatures for summary plots, K ($^{\circ}$ F)
TSP		GETIN	input pin-fin spacing, cm (in.)
TTEMP		PLNUM	variable to save temperature value, $^{\circ}$ R

Variable	Common	Subroutine	Definition
TTEXT		PLNUM	coolant total temperature in plenum, $^{\circ}\text{F}$
TTIN		NTTACT	coolant-supply temperature, $^{\circ}\text{F}$, at entrance to coolant plenum for given time
TTIO(50)	BOUND		input table of coolant-supply temperature, K ($^{\circ}\text{F}$), as function of time
TTIP(80)	BOUND		table of blade tip temperature, $^{\circ}\text{F}$, for given time
TTIPIN(400)	BOUND		input table of blade tip temperature, K ($^{\circ}\text{F}$), as function of time
TTOTC(80)		TCOEF	coolant total temperature at each station for given slice, $^{\circ}\text{F}$
TTX		PLNUM	coolant total temperature at inlet to coolant plenum for given slice, $^{\circ}\text{R}$
TTYME		WROUT	current time in transient, sec
TT1(20)		PLNUM	total-temperature distribution, $^{\circ}\text{F}$, in coolant plenum
TW		THRCON	midwall temperature for evaluating thermal conductivity
TXN		GETIN	input spanwise spacing of impingement jets at given station, cm (in.)
TXX		PLNUM	average static temperature in coolant plenum for given slice, $^{\circ}\text{F}$
TYME	TRNSNT		time in transient calculations, sec
TYMMAX	TRNSNT		maximum time to which transient is carried, sec
T0G	TCO		impingement-jet temperature for given slice, $^{\circ}\text{R}$
T1		PLNUM	inlet total temperature in coolant plenum for given slice, $^{\circ}\text{R}$
UA(2)		INPRT	alphanumeric array containing area units

Variable	Common	Subroutine	Definition
UL(2)		INPRT	alphanumeric array containing length units
V		TCOEF	factor to accelerate convergence of pressure iterations
VARIB(15)		PLOTMF	alphanumeric array containing some plot labels
VAR(12)		PLOTMF	array containing plotting controls
VDP		HCPINS	pin-fin diameter at given station, in.
VOLBAR		TARRAY	coolant-channel volume element at entrance to trailing-edge region, in ³
VSP		HCPINS	pin-fin spacing at given station, in.
V1		PLNUM	intermediate term in coolant-plenum calculations
W(15)	RADL		coolant flow-rate at entrance to each slice, lbm/hr
WC		HCOOL	absolute value of coolant flow-rate, lbm/sec
WCHK(80)	CHKHOL		alphanumeric variable used to indicate choked flow in impingement jets
WCHKDM	CHKHOL		alphanumeric variable used to indicate choked flow in holes dumping coolant to trailing-edge region
WCHOKE		PLNUM	coolant flow-rate at entrance to given slice, lbm/hr
WCR		FLows	critical flow-rate, lbm/sec
WCROS(2, 15, 80)	TCO		coolant-channel crossflow rate at each station for each slice for two time steps, lbm/sec
WDUMP	TCO		rate of coolant flow being dumped directly from plenum to trailing-edge region, lbm/sec
WFC(80)	TCO		film-cooling flow rate at each station for given slice, lbm/sec

Variable	Common	Subroutine	Definition
WFCDUM		FLAWS	intermediate variable in film-cooling flow rate calculation
WFCDUM		PARRAY	total film-cooling flow rate at given station, lbm/sec
WIM	TCO		total impingement-jet flow rate for given slice, lbm/sec
WJ(15,80)	TCO		impingement-jet flow rate for each station for each slice, lbm/sec
WPLEN	BOUND		input initial guess at total coolant flow, kg/hr (lbm/hr)
WPLENC		INPRT	total coolant flow in SI units, kg/hr
WPLENO		NTTACT	variable to save previous estimate of total coolant flow, lbm/hr
WS	RADL		rotor speed at given time, rpm
WSVST(50)	BOUND		input table of rotor speed, rpm, versus time, sec
WUSED		NTTACT	cumulative amount of coolant used, up to current slice, lbm/hr
WXCP		TARRAY	coolant flow-rate times specific-heat term
WXX		PLNUM	coolant flow-rate at entrance to given slice, lbm/hr
X(80)		TCOEF	coolant-channel node locations, in., used to set initial pressure distribution
XBAR		FLAWS	term in film-cooling effectiveness correlation
XCC		INPRT	coolant-channel distance from station 1 to given station, cm (in.)
XDUM		FLAWS	dummy variable used to save location of film-cooling hole

Variable	Common	Subroutine	Definition
XFC(80)	FLMCOL		distance from station with film-cooling holes to downstream stations without film-cooling holes, in.
XIS		INPRT	inside-wall surface distance from station 1 to given station
XJN		INPRT	distance along junction of cladding and wall metal from station 1 to given station, cm (in.)
XK(4)		PLNUM	pressure change in coolant plenum
XL		HCOOL	length of inner surface used in leading-edge impingement correlation, in.
XL(4)		PLNUM	temperature change in coolant plenum
XLABL(29)		PLOTMF	alphameric array of plot labels
XLABL2(15)		PLOTMF	alphameric array of plot labels
XLBL(20)		PLOTMF	array of x-coordinates to be plotted as slice numbers
XMM		INPRT	midwall distance from station 1 to given station, cm (in.)
XMU		FLOWS GASTBL HCFRCD HCOOL HCPINS PLNUM TARRAY	viscosity, lbm/ft · hr
XMUC(80)	FLMCOL		coolant viscosity, lbm/ft · hr, at each station, evaluated at mean temperature between inner-wall surface and bulk coolant temperatures
XMUM		FLOWS HCOOL	hot-gas viscosity, lbm/ft · hr
XMUTOG		HCOOL	coolant viscosity based on coolant-supply temperature, lbm/ft · hr

Variable	Common	Subroutine	Definition
XN(80)	TCO		spanwise spacing of impingement holes at each station, in.
XNN		PLNUM	factor for increasing number of stations in coolant plenum
XOD		PREP	ratio of impingement-hole spacing to hydraulic diameter, at given station
XOS		INPRT	wall outer-surface distance from station 1 to given station, cm (in.)
XOVERD		HCOOL	ratio of impingement-hole spacing to hydraulic diameter, at given station
XOVR		HCPINS	location of zero temperature gradient in pin fins
XP		HCOOL	length of pressure-side inner-wall surface in leading-edge impingement region
XP(80)		PLOTMF	pressure-side, dimensionless distance along midwall plane from station 1 to each station
XP		PREP	distance of given pressure-side station from station 1, in.
XPF		PREP	interpolating fraction in BCXP table
XPL		PLOTMF	overall length along pressure-side, midwall plane, cm (in.)
XS		HCOOL	length of suction-side, inner-wall surface in leading-edge impingement region
XS(80)		PLOTMF	suction-side, dimensionless distance along midwall plane from station 1 to each station
XS		PREP	distance of given suction-side station from station 1, in.
XSF		PREP	interpolating function in BCXS table

Variable	Common	Subroutine	Definition
XSL		PLOTMF	overall length along suction-side, mid-wall plane, cm (in.)
XTEST		PLNUM	convergence test variable
XTOT		WROUT	overall outer-surface length around blade, in.
XTOTMD		WROUT	overall midwall length around blade, in.
XXN		PLNUM	number of spanwise stations per slice in coolant plenum
Y(320)		PLOTMF	array containing temperature values to be plotted
YCNVUU		TARRAY	indicates forced-convection heat transfer at last forward-region station on pressure side
YCONV		TARRAY	indicates forced-convection heat transfer at given station
YCONVU		TARRAY	indicates forced-convection heat transfer at station immediately upstream of given station
YFINS		TARRAY	indicates pin-fin heat transfer at given station
YFINSU		TARRAY	indicates pin-fin heat transfer at station immediately upstream of given station
YFNSUU		TARRAY	indicates pin-fin heat transfer at last forward-region station on pressure side
YIMP		TARRAY	indicates impingement heat transfer at given station
YIMPU		TARRAY	indicates impingement heat transfer at station immediately upstream of given station
YIMPUU		TARRAY	indicates impingement heat transfer at last forward-region station on pressure side

Variable	Common	Subroutine	Definition
YLABL(7)		PLOTMF	alphanumeric array for labeling plots
YLABL2(11)		PLOTMF	alphanumeric array for labeling plots
YLBL(20)		PLOTMF	array of coordinates of points to be plotted as slice numbers
YMAX		PLOTMF	maximum value of y-coordinates on plot
YMIN		PLOTMF	minimum value of y-coordinates on plot
YPLABL(10)		PLOTMF	alphanumeric array for labeling plots
YTEM(80)		PLOTMF	array to be plotted
ZED		PLNUM	coolant-plenum pressure-drop parameter for given slice
ZOVERD		HCOOL	ratio of coolant-channel width to impingement-hole hydraulic diameter
Z1(15)		PLNUM	coolant-plenum pressure-drop parameter for each slice
Z3		PLNUM	intermediate term involving coolant flow
Z4		PLNUM	intermediate term involving coolant flow

PROGRAM LISTING

```

C-----SOURCE.NTTACT---THIS IS THE MAIN PROGRAM. BLOCK DATA SUBPROGRAM
C                                     GASDAT MUST BE LOADED FIRST.
C                                     NTTACT--0001
C                                     NTTACT--0002
C                                     NTTACT--0003
C      TRANSIENT THERMAL ANALYSIS OF A COOLED TURBINE BLADE
C      *               *               *               *               *
C      TACT1
C      NTTACT--0004
C      NTTACT--0005
C      NTTACT--0006
C      NTTACT--0007
C      COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000),
Z      BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),
Z      BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),
Z      QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80),
Z      QTIPIN(400), QTIP(80), RHOVG(400), PEX(400),
Z      BCTIME(50), TTIO(50), PTIO(50), WPLEN,
Z      MSVST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCP
C      NTTACT--0008
C      COMMON /FLMCOL/ RHOVGA(80), FG(80), XPC(80), FLMEFF(80),
Z      XMUC(80), EMES(80), REFC(80), NFCSUP(80)
C      NTTACT--0009
C      NTTACT--0010
C      COMMON /GAAS/ GS(200), NG
C      NTTACT--0011
C      COMMON /RADL/ APLN(15), DPLN(15), RIN(15), ROUT(15),
Z      PIN(15), TIN(15), W(15), WS
C      NTTACT--0012
C      COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000),
Z      IPLOT, MD1, MD2, MD3, IADJIN, IWRITE
C      NTTACT--0013
C      COMMON /TCO/ ADUMP, BTA, CD, CP,
Z      GAM, PIM, R, SPAN, TOG,
Z      WDUMP, WIM, AKC(15,80), AKW(15,80),
Z      A(400), AJET(80), AM2(80), CNUM(80),
Z      DH(80), DHF(80), DHJ(80),
Z      DLX(400), FF(80), HC(80), HG(80),
Z      P(2,15,80), PEXIT(15), PUMP(80), QG(80),
Z      QSNK(80), RR(80), S(15), T(2,15,400),
Z      TG(80), TAU(400), WFC(80),
Z      WJ(15,80), WCROS(2,15,80), XN(80),
Z      ICOR, IFILM, IHUB, ITIP,
Z      ISBLOK, ISLICE, NBLKSZ, NSLICE,
Z      NPWD, NSTA, IHC(80)
C      NTTACT--0014
C      NTTACT--0015
C      NTTACT--0016
C      NTTACT--0017
C      COMMON /TRNSNT/ RHOC, RHOM, SPHTC, SPHTM,
Z      DLTME, TYME, TEPS, TYMMAX
C      NTTACT--0018
C      COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSFL(2),
Z      CTMPF(2), CTCON(2), CDEN(2), CSPHT(2), CGASC(2),
Z      CVISC(2), CRHOVG(2), IUNITS
C      NTTACT--0019
C      DIMENSION DP(80), SP(80), ALPH(12), ALPH2(4), CD1(200)
C      NTTACT--0020
C      TTIO = TOTAL TEMPERATURE OF BLADE COOLING AIR AT INLET
C      WPLEN = ESTIMATE OF COOLANT FLOW RATE - USED AS FIRST GUESS
C      PTIO = TOTAL PRESSURE OF BLADE COOLING AIR AT INLET
C      PEX = EXTERNAL GAS STREAM STATIC PRESSURE AT TRAILING EDGE
C      NTTACT--0021
C      DATA ALPH/' THI','S JO','B WA','S ST','ARTE','D AT',
Z      ' ',' ',' ','ON ',' ',' ',' '/
C      DATA NCHAR/16/
C      NTTACT--0022
C      MD1 = 0
C      MD2 = 0
C      NTTACT--0023
C      NTTACT--0024
C      NTTACT--0025
C      NTTACT--0026
C      NTTACT--0027
C      NTTACT--0028
C      NTTACT--0029
C      NTTACT--0030
C      NTTACT--0031
C      NTTACT--0032
C      NTTACT--0033
C      NTTACT--0034
C      NTTACT--0035
C      NTTACT--0036
C      NTTACT--0037
C      NTTACT--0038
C      NTTACT--0039
C      NTTACT--0040
C      NTTACT--0041
C      NTTACT--0042
C      NTTACT--0043
C      NTTACT--0044
C      NTTACT--0045
C      NTTACT--0046
C      NTTACT--0047
C      NTTACT--0048
C      NTTACT--0049
C      NTTACT--0050
C      NTTACT--0051
C      NTTACT--0052
C      NTTACT--0053
C      NTTACT--0054
C      NTTACT--0055
C      NTTACT--0056
C      NTTACT--0057
C      NTTACT--0058
C      NTTACT--0059
C      NTTACT--0060

```

TIME = -1.0	NTTACT--0061
DLTIME = 0.0	NTTACT--0062
C	NTTACT--0063
C TO GET AN ABBREVIATED OUTPUT OF MID-WALL TEMPERATURES AT THE TERMINAL	NTTACT--0064
C FOR EACH SLICE, ENTER:	NTTACT--0065
C AT TACT.50;SET TACT.MD1=1	NTTACT--0066
C	NTTACT--0067
MD3 = 0	NTTACT--0068
K = 1	NTTACT--0069
C	NTTACT--0070
C RECORD STARTING TIME, TO BE USED TO IDENTIFY MICROFILM PLOTS	NTTACT--0071
C	NTTACT--0072
CALL TIME(NCHAR,ALPH2)	NTTACT--0073
ALPH(7) = ALPH2(3)	NTTACT--0074
ALPH(8) = ALPH2(4)	NTTACT--0075
ALPH(10) = ALPH2(1)	NTTACT--0076
ALPH(11) = ALPH2(2)	NTTACT--0077
WRITE(6,425) (ALPH(I),I=1,12)	NTTACT--0078
WRITE(8,425) (ALPH(I),I=1,12)	NTTACT--0079
C	NTTACT--0080
C READ IN DATA	NTTACT--0081
C	NTTACT--0082
CALL GETIN(IWRITE,TYHMAX,WSVST,IADJIN)	NTTACT--0083
C	NTTACT--0084
C WRITE TITLE PAGE	NTTACT--0085
C	NTTACT--0086
WRITE(6,400)	NTTACT--0087
WRITE(6,425) (ALPH(I),I=1,12)	NTTACT--0088
WRITE(6,430) (TITLE(I),I=1,30)	NTTACT--0089
400 FORMAT(1H1,////////,50X,'***** OUTPUT *****',////////)	NTTACT--0090
425 FORMAT(/36X,12A4)	NTTACT--0091
430 FORMAT(/1X,30A4)	NTTACT--0092
C	NTTACT--0093
TTIN = TTIO(1)	NTTACT--0094
PTIN = PTIO(1)	NTTACT--0095
440 WPLENO = WPLEN	NTTACT--0096
PTNOLD = PTIN	NTTACT--0097
PTIO(1) = PTIN	NTTACT--0098
TYME = 0.0	NTTACT--0099
NTYM = 1	NTTACT--0100
IF (DLTIME.GT.0.) NTYM = TYHMAX/DLTIME + 1	NTTACT--0101
MODST = 5*NSTA	NTTACT--0102
C	NTTACT--0103
C	NTTACT--0104
C START MARCHING	NTTACT--0105
C	NTTACT--0106
DO 1100 ITYM = 1,NTYM	NTTACT--0107
ITYME = ITYM-1	NTTACT--0108
NTTG = ITYM	NTTACT--0109
TYME = ITYME*DLTIME	NTTACT--0110
IF (ITYM.EQ.1) TYME = -1.	NTTACT--0111
C	NTTACT--0112
C	NTTACT--0113
C	NTTACT--0114
C-- EVALUATE TIME DEPENDENT BOUNDARY CONDITIONS -----	NTTACT--0115
C -	NTTACT--0116
C	NTTACT--0117
PTIN = PTIO(1)	NTTACT--0118
IF (TYME.LT.0.0) GO TO 490	NTTACT--0119
C	NTTACT--0120

C-- LOCATE COOLANT SUPPLY PRESSURE FOR TIME	NTTACT--0121
C	NTTACT--0122
DO 450 I = 4,50,2	NTTACT--0123
PTIN = PTIO(I-3)	NTTACT--0124
IF (PTIO(I).LE.0.0) GO TO 460	NTTACT--0125
IPTIO = I-1	NTTACT--0126
IF (TYME.LE.PTIO(I).AND.TYME.GT.PTIO(I-2)) GO TO 455	NTTACT--0127
450 CONTINUE	NTTACT--0128
455 DENOM = PTIO(IPTIO+1)-PTIO(IPTIO-1)	NTTACT--0129
IF (DENOM.GT.0.) PTIN = PTIO(IPTIO-2) +	NTTACT--0130
Z (PTIO(IPTIO)-PTIO(IPTIO-2)) * (TYME-PTIO(IPTIO-1)) /DENOM	NTTACT--0131
460 CONTINUE	NTTACT--0132
C	NTTACT--0133
C-- LOCATE COOLANT SUPPLY TEMPERATURE FOR TIME	NTTACT--0134
C	NTTACT--0135
DO 470 I = 4,50,2	NTTACT--0136
TTIN = TTIO(I-3)	NTTACT--0137
IF (TTIO(I).LE.0.0) GO TO 490	NTTACT--0138
ITTIO = I-1	NTTACT--0139
IF (TYME.LE.TTIO(I).AND.TYME.GT.TTIO(I-2)) GO TO 475	NTTACT--0140
470 CONTINUE	NTTACT--0141
475 DENOM = TTIO(ITTIO+1)-TTIO(ITTIO-1)	NTTACT--0142
IF (DENOM.GT.0.) TTIN = TTIO(ITTIO-2) +	NTTACT--0143
Z (TTIO(ITTIO)-TTIO(ITTIO-2)) * (TYME-TTIO(ITTIO-1)) /DENOM	NTTACT--0144
490 CONTINUE	NTTACT--0145
C	NTTACT--0146
C LOCATE THE VALUE OF THE WHEEL SPEED FOR THE CURRENT TIME.	NTTACT--0147
C	NTTACT--0148
WS = WSVST(1)	NTTACT--0149
IF (TYME.LE.0.0) GO TO 530	NTTACT--0150
DO 510 I = 4,50,2	NTTACT--0151
WS = WSVST(I-3)	NTTACT--0152
IF (WSVST(I).LE.0.0) GO TO 530	NTTACT--0153
IWS = I-1	NTTACT--0154
IF (TYME.LE.WSVST(I).AND.TYME.GT.WSVST(I-2)) GO TO 520	NTTACT--0155
510 CONTINUE	NTTACT--0156
520 DENOM = WSVST(IWS+1) - WSVST(IWS-1)	NTTACT--0157
IF (DENOM.GT.0.0) WS = WSVST(IWS-2) +	NTTACT--0158
Z (WSVST(IWS)-WSVST(IWS-2)) * (TYME-WSVST(IWS-1)) /DENOM	NTTACT--0159
530 CONTINUE	NTTACT--0160
C	NTTACT--0161
C	NTTACT--0162
C LOCATE THE VALUE FOR PEXIT, GAS STATIC PRESSURE AT EXIT OF BLADE,	NTTACT--0163
C FOR THE CURRENT TIME AND ALL SLICES.	NTTACT--0164
C	NTTACT--0165
IF (TYME.GT.0.0) GO TO 533	NTTACT--0166
DO 532 I = 1,NSLICE	NTTACT--0167
PEXIT(I) = PEX(I)	NTTACT--0168
IF (PEX(I).LE.0.0) PEXIT(I) = PEX(1)	NTTACT--0169
532 CONTINUE	NTTACT--0170
533 CONTINUE	NTTACT--0171
IF (BCTIME(2).LE.0.0) GO TO 545	NTTACT--0172
DO 535 I = 2,50	NTTACT--0173
IPEX = I-1	NTTACT--0174
IF (TYME.LE.BCTIME(I).AND.TYME.GT.BCTIME(I-1)) GO TO 540	NTTACT--0175
535 CONTINUE	NTTACT--0176
540 DENOM = BCTIME(IPEX+1) - BCTIME(IPEX)	NTTACT--0177
IF (DENOM.EQ.0.0) GO TO 545	NTTACT--0178
TYMPRC = (TYME - BCTIME(IPEX)) /DENOM	NTTACT--0179
DO 542 I = 1,NSLICE	NTTACT--0180

	IC = (IPEX-1)*NSLICE + .I	NTTACT--0181
	PEXIT(I) = PEX(IC) + (PEX(IC+NSLICE)-PEX(IC))*TYMFR	NTTACT--0182
542	CONTINUE	NTTACT--0183
	IF (IFILM.NE.2) GO TO 545	NTTACT--0184
C		NTTACT--0185
C--	SET INTERPOLATED VALUES OF FREE STREAM RHO*V FOR THIS TIME	NTTACT--0186
C		NTTACT--0187
	DO 543 I = 1,NSTA	NTTACT--0188
	IRO = (IPEX-1)*NSTA + I	NTTACT--0189
	IRN = IPEX*NSTA + I	NTTACT--0190
543	RHOVGA(I) = RHOVG(IRO) + TYMFR*(RHOVG(IRN)-RHOVG(IRO))	NTTACT--0191
C		NTTACT--0192
C		NTTACT--0193
545	CONTINUE	NTTACT--0194
C		NTTACT--0195
C		NTTACT--0196
C--	SET TIME INTERPOLATED VALUES OF QHUB OR THUB AND QTIP OR TTIP.	NTTACT--0197
	IF (BCTIME(2).LE.0.0) GO TO 555	NTTACT--0198
	DO 550 I = 1,NSTA	NTTACT--0199
	IQO = (IPEX-1)*NSTA + I	NTTACT--0200
	IQN = IPEX*NSTA + I	NTTACT--0201
	IF (IHUB.EQ.1) THUB(I) = THUBIN(IQO) +	NTTACT--0202
Z	TYMFR*(THUBIN(IQN)-THUBIN(IQO))	NTTACT--0203
	IF (IHUB.EQ.3) QHUB(I) = QHUBIN(IQO) +	NTTACT--0204
Z	TYMFR*(QHUBIN(IQN)-QHUBIN(IQO))	NTTACT--0205
	IF (ITIP.EQ.1) TTIP(I) = TTIPIN(IQO) +	NTTACT--0206
Z	TYMFR*(TTIPIN(IQN)-TTIPIN(IQO))	NTTACT--0207
	IF (ITIP.EQ.3) QTIP(I) = QTIPIN(IQO) +	NTTACT--0208
Z	TYMFR*(QTIPIN(IQN)-QTIPIN(IQO))	NTTACT--0209
550	CONTINUE	NTTACT--0210
	GO TO 565	NTTACT--0211
C		NTTACT--0212
555	CONTINUE	NTTACT--0213
	DO 560 I = 1,NSTA	NTTACT--0214
	IF (IHUB.EQ.1) THUB(I) = THUBIN(I)	NTTACT--0215
	IF (IHUB.EQ.3) QHUB(I) = QHUBIN(I)	NTTACT--0216
	IF (ITIP.EQ.1) TTIP(I) = TTIPIN(I)	NTTACT--0217
	IF (ITIP.EQ.3) QTIP(I) = QTIPIN(I)	NTTACT--0218
560	CONTINUE	NTTACT--0219
C		NTTACT--0220
565	TCIN = TTIN	NTTACT--0221
	IF (ITYM.GT.1) WPLEN = WUSED*PTIN/PTNOLD	NTTACT--0222
570	WPLENO = WPLEN	NTTACT--0223
	PTNOLD = PTIN	NTTACT--0224
	WUSED = 0.0	NTTACT--0225
C		NTTACT--0226
C	CALCULATE TEMPERATURE AND PRESSURES FOR EACH SLICE OF THE BLADE	NTTACT--0227
C		NTTACT--0228
	DO 1000 I = 1,NSLICE	NTTACT--0229
	ISLICE = I	NTTACT--0230
C		NTTACT--0231
C	FIRST DETERMINE IMPINGEMENT PLENUM CONDITIONS	NTTACT--0232
C		NTTACT--0233
	CALL PLNUM(WPLEN,PPLEN,PTIN,TPLEN,TCIN)	NTTACT--0234
	TOG = TPLEN + 460.	NTTACT--0235
	PIM = PPLEN	NTTACT--0236
	IF (IUNITS.EQ.1) GO TO 860	NTTACT--0237
	WRITE(6,800) I,PIM,TOG	NTTACT--0238
800	FORMAT(1H2//10X,100('*'))//30X,'THE IMPINGEMENT PLENUM CONDITIONS'	NTTACT--0239
Z	' FOR SLICE NO.',I2,' ARE: '//60X,'PIM = ',F7.2,	NTTACT--0240

Z	' PSIA, '//60X, 'TOG = ', F7.2, ' DEG. R'//10X, 100('*)	NTTACT--0241
C		NTTACT--0242
	IF (I.EQ.1) WRITE(6,850) WPLEN	NTTACT--0243
850	FORMAT(1H ,//30X, 'CENTRAL PLENUM FLOW IS ', F6.1, ' LBM/HR', //)	NTTACT--0244
	GO TO 890	NTTACT--0245
C		NTTACT--0246
860	CTOG = TOG/1.8	NTTACT--0247
	CPIM = PIM/CPRSR(1)	NTTACT--0248
	WRITE(6,870) I, CPIM, CTOG	NTTACT--0249
870	FORMAT(1H2//10X, 100('*)//30X, 'THE IMPINGEMENT PLENUM CONDITIONS'	NTTACT--0250
Z	' FOR SLICE NO.', I2, ' ARE: '//60X, 'PIM = ', F7.2,	NTTACT--0251
Z	' KPA, '//60X, 'TOG = ', F7.2, ' K '//10X, 100('*)	NTTACT--0252
	CWPLEN = WPLEN/CMSFL(1)	NTTACT--0253
	IF (I.EQ.1) WRITE(6,880) CWPLEN	NTTACT--0254
880	FORMAT(1H ,//30X, 'CENTRAL PLENUM FLOW IS ', F6.1, ' KG/HR', //)	NTTACT--0255
C		NTTACT--0256
990	CONTINUE	NTTACT--0257
C		NTTACT--0258
C	THEN COMPUTE TEMPERATURES AND PRESSURES	NTTACT--0259
C		NTTACT--0260
	CALL PREP(I, NTTG)	NTTACT--0261
C		NTTACT--0262
	CALL TCOEF(IWRITE, WS, K, IPLOT, ALPH2)	NTTACT--0263
	IF (IPLOT.GT.0) CALL PLOTMF (ALPH2)	NTTACT--0264
	IF (MD1.EQ.0) GO TO 975	NTTACT--0265
C		NTTACT--0266
C	THIS BLOCK PRINTS SPECIAL CONDENSED OUTPUT TO THE TERMINAL IF MD1 > 0	NTTACT--0267
C		NTTACT--0268
	IC1 = 0	NTTACT--0269
	DO 955 II = 1, NSTA, 2	NTTACT--0270
	IC1 = IC1 + 1	NTTACT--0271
	NMW = 5*II - 4	NTTACT--0272
955	CD1(IC1) = T(2, I, NMW) - 460.	NTTACT--0273
	DO 960 II = 2, NSTA, 2	NTTACT--0274
	IC1 = IC1 + 1	NTTACT--0275
	NMW = 5*II - 4	NTTACT--0276
960	CD1(IC1) = T(2, I, NMW) - 460.	NTTACT--0277
	WRITE(8,962) I, K	NTTACT--0278
962	FORMAT(/' BLADE SLICE ', I2, ', OVERALL FLOW LOOP ', I2,	NTTACT--0279
Z	', SURFACE TEMPERATURE, (F), STARTING AT LEADING EDGE'//)	NTTACT--0280
	INUM = NSTA/2 + 1	NTTACT--0281
	WRITE(8,964) (CD1(II), II=1, INUM)	NTTACT--0282
964	FORMAT(' PRESSURE SIDE'/10 (2X, F7.1))	NTTACT--0283
	INUM = INUM + 1	NTTACT--0284
	WRITE(8,966) CD1(1), (CD1(II), II=INUM, NSTA)	NTTACT--0285
966	FORMAT(/' SUCTION SIDE'/10 (2X, F7.1))	NTTACT--0286
C		NTTACT--0287
C		NTTACT--0288
975	CONTINUE	NTTACT--0289
C		NTTACT--0290
C	CHECK HOW MUCH PLENUM FLOW IS LEFT	NTTACT--0291
C		NTTACT--0292
	WUSED = WUSED + 3600.*WIM	NTTACT--0293
	EXCESW = WPLEN - 3600.*WIM	NTTACT--0294
	IF (EXCESW.GT.0..AND.I.LT.NSLICE) WPLEN = EXCESW	NTTACT--0295
	IF (EXCESW.LT.0..AND.I.LT.NSLICE) WRITE(6,985) I	NTTACT--0296
985	FORMAT(/' RAN OUT OF MASS FLOW IN BRANCH NO. ', I2)	NTTACT--0297
1000	CONTINUE	NTTACT--0298
C		NTTACT--0299
C		NTTACT--0300

IF (IUNITS.EQ.2) WRITE(6,1010) WUSED	NTTACT--0301
1010 FORMAT(/30X,'AMOUNT OF COOLANT ACTUALLY USED AT THIS TIME'	NTTACT--0302
Z ' STEP IS ',F6.1,' LBM/HR')	NTTACT--0303
CWUSED = WUSED/CMSPL(1)	NTTACT--0304
IF (IUNITS.EQ.1) WRITE(6,1011) CWUSED	NTTACT--0305
1011 FORMAT(/30X,'AMOUNT OF COOLANT ACTUALLY USED AT THIS TIME'	NTTACT--0306
Z ' STEP IS ',F6.1,' KG/HR')	NTTACT--0307
C	NTTACT--0308
DO 1040 I = 1,NSLICE	NTTACT--0309
DO 1020 J = 1,NODST	NTTACT--0310
1020 T(1,I,J) = T(2,I,J)	NTTACT--0311
DO 1040 J = 1,NSTA	NTTACT--0312
1040 P(1,ISLICE,J) = P(2,ISLICE,J)	NTTACT--0313
IF (TYME.GT.0.0) GO TO 1100	NTTACT--0314
C	NTTACT--0315
C---> ADJUST COOLANT FLOW AND RECALCULATE TEMPERATURES, ETC. FOR STEADY	NTTACT--0316
C STATE CASE OR TIME =0.0	NTTACT--0317
C	NTTACT--0318
EXCESW = WPLENO - WUSED	NTTACT--0319
IF (ABS(EXCESW).LT..01*WPLENO) GO TO 1100	NTTACT--0320
IF (IADJIN.GT.0) GO TO 1050	NTTACT--0321
C	NTTACT--0322
C ---> NORMAL ADJUSTMENT IS ON WPLEN	NTTACT--0323
C	NTTACT--0324
WPLEN = WPLENO - .995*EXCESW	NTTACT--0325
K = K + 1	NTTACT--0326
PTIN = PTNOLD	NTTACT--0327
TCIN = TTIN	NTTACT--0328
GO TO 570	NTTACT--0329
C	NTTACT--0330
C ----> SPECIAL CASE, FOR IADJPT > 0, ADJUSTMENT IS ON PTIN	NTTACT--0331
C	NTTACT--0332
1050 PTIN = PEXIT(1) + (PTNOLD-PEXIT(1))*(WPLENO/WUSED)**1.6	NTTACT--0333
WPLEN = WPLENO	NTTACT--0334
TCIN = TTIN	NTTACT--0335
K = K + 1	NTTACT--0336
GO TO 570	NTTACT--0337
C	NTTACT--0338
1100 CONTINUE	NTTACT--0339
C	NTTACT--0340
C	NTTACT--0341
IF (IUNITS.EQ.1) GO TO 3860	NTTACT--0342
3000 WRITE(6,3500) K,WPLENO	NTTACT--0343
3500 FORMAT(/,20X,80(' '),/,23X,12,' LOOP(S) ON INITIAL COOLANT FLOW'	NTTACT--0344
Z ' WERE USED. FINAL VALUE IS ',F7.2,' LB/HR')	NTTACT--0345
WRITE(6,3600) EXCESW	NTTACT--0346
3600 FORMAT(5X,'RESIDUAL COOLING AIR FLOW IS ',F9.4,' LBM/HR',/	NTTACT--0347
Z 20X,80(' '))	NTTACT--0348
WRITE(6,425) (ALPH(I),I=1,12)	NTTACT--0349
C	NTTACT--0350
GO TO 3900	NTTACT--0351
C	NTTACT--0352
3860 CWPLEN = WPLENO/CMSPL(1)	NTTACT--0353
CEXCSW = EXCESW/CMSPL(1)	NTTACT--0354
WRITE(6,3870) K,CWPLEN	NTTACT--0355
3870 FORMAT(/,20X,80(' '),/,23X,12,' LOOP(S) ON INITIAL COOLANT FLOW'	NTTACT--0356
Z ' WERE USED. FINAL VALUE IS ',F7.2,' KG/HR')	NTTACT--0357
WRITE(6,3880) CEXCSW	NTTACT--0358
3880 FORMAT(5X,'RESIDUAL COOLING AIR FLOW IS ',F9.4,' KG/HR',/	NTTACT--0359
Z 20X,80(' '))	NTTACT--0360

C	WRITE(6,425) (ALPH(I),I=1,12)	NTTACT--0361
C		NTTACT--0362
3900	CONTINUE	NTTACT--0363
	MD2 = 1	NTTACT--0364
	IF (NSLICE.GT.1.AND.IPLOT.GT.0) CALL PLOTMF(ALPH2)	NTTACT--0365
	WRITE(6,425) (ALPH(I),I=1,12)	NTTACT--0366
	STOP	NTTACT--0367
	END	NTTACT--0368
C----	SOURCE.NFLOEST	NFLOEST 0369
	SUBROUTINE FLOWS (JS,DELTAN,ICHOKE,AMCHOK)	NFLOEST 0370
C		NFLOEST 0371
C=====		NFLOEST 0372
C		NFLOEST 0373
C-	SOURCE.NFLOEST--	NFLOEST 0374
C		NFLOEST 0375
C	THIS SUBROUTINE COMPUTES THE FOLLOWING---	NFLOEST 0376
C	WJ, IMPINGEMENT JET FLOW RATES (LBM/SEC).	NFLOEST 0377
C	WCROS, THE CHANNEL CROSSFLOWS (LBM/SEC).	NFLOEST 0378
C	AM2, THE SQUARE OF THE CHANNEL MACH NUMBER.	NFLOEST 0379
C	WDUMP, A FLOWRATE DUMPED DIRECTLY FROM CENTRAL PLENUM INTO TRAILING	NFLOEST 0380
C	EDGE CHANNEL (LBM/SEC).	NFLOEST 0381
C	WIM, THE TOTAL COOLANT FLOW USED FOR THIS SLICE (LBM/SEC).	NFLOEST 0382
C	WPC, FILM COOLING HOLE FLOW RATES (LBM/SEC).	NFLOEST 0383
C	FP, THE CHANNEL FRICTION FACTOR, EITHER FOR LAMINAR, TURBULENT, OR	NFLOEST 0384
C	A PIN FIN ARRAY.	NFLOEST 0385
C	PLMEFF, A FILM COOLING EFFECTIVENESS.	NFLOEST 0386
C		NFLOEST 0387
C=====		NFLOEST 0388
C	COMMON /CHKHOL/ WCHK(80), WCHKDM	NFLOEST 0389
C		NFLOEST 0390
	COMMON /FLMCOL/ RHOVGA(80), PG(80), XPC(80), PLMEFF(80),	NFLOEST 0391
Z	XMUC(80), ENES(80), REFC(80), NFCSUP(80)	NFLOEST 0392
C		NFLOEST 0393
	COMMON /FRIC/ ALPHA, BETA, DELTA, EPS	NFLOEST 0394
C		NFLOEST 0395
	COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),	NFLOEST 0396
Z	CPC(80), GAMC(80), DUMR1(80), DUMR2(80)	NFLOEST 0397
C		NFLOEST 0398
	COMMON /TCO/ ADUMP, BTA, CD, CP, NFLOEST 0399	
Z	GAM, PIN, R, SPAN, TOG, NFLOEST 0400	
Z	WDUMP, WIM, AKC(15,80), AKW(15,80), NFLOEST 0401	
Z	A(400), AJET(80), AM2(80), CNUM(80), NFLOEST 0402	
Z	DH(80), DHP(80), DHJ(80), NFLOEST 0403	
Z	DLX(400), FF(80), HC(80), HG(80), NFLOEST 0404	
Z	P(2,15,80), PEXIT(15), PUMP(80), QG(80), NFLOEST 0405	
Z	QSNK(80), RR(80), S(15), T(2,15,400), NFLOEST 0406	
Z	TG(80), TAU(400), WPC(80), NFLOEST 0407	
Z	WJ(15,80), WCROS(2,15,80), XN(80), NFLOEST 0408	
Z	ICOR, IFILM, IHUB, ITIP, NFLOEST 0409	
Z	ISBLOK, ISLICE, NBLKSZ, NSLICE, NFLOEST 0410	
Z	NPWD, NSTA, IHC(80), NFLOEST 0411	
C		NFLOEST 0412
	COMMON /TRNSNT/ RHOC, RHOM, SPHTC, SPHTM, NFLOEST 0413	
Z	DLTYME, TYME, TEPS, TYMMAX NFLOEST 0414	
C		NFLOEST 0415
	DATA CHKD/'@'/, UNCHKD/' '/	NFLOEST 0416
C		NFLOEST 0417
	DIMENSION DELTAN(15)	NFLOEST 0418
C		NFLOEST 0419
		NFLOEST 0420

C	FOLLOWING VARIABLES NEEDED TO CALCULATE FILM COOLING EFFECTIVENESS.	NFLOEST 0421
C	THEY ARE TRANSMITTED THROUGH COMMON FLMCOL	NFLOEST 0422
C	WHERE-	NFLOEST 0423
C	PG IS GAS SIDE STATIC PRESSURE DISTRIBUTION; XFC IS THE	NFLOEST 0424
C	DISTANCE A STATION IS DOWNSTREAM FROM THE NEAREST	NFLOEST 0425
C	ROW OF FILM COOLING HOLES, (IN); FLMEFF IS THE CALCULATED	NFLOEST 0426
C	FILM EFFECTIVENESS AT EACH STATION;	NFLOEST 0427
C	XMUC IS COOLANT VISCOSITY BASED ON LOCAL COOLANT TEMPERATURE;	NFLOEST 0428
C	EMES IS THE PRODUCT M*S, WHERE M IS THE BLOWING	NFLOEST 0429
C	RATIO, AND S IS AN EQUIVALENT SLOT WIDTH; REFC IS THE FILM	NFLOEST 0430
C	REYNOLDS NUMBER, BASED ON S; AND NFCSUP IDENTIFIES	NFLOEST 0431
C	THE STATION NUMBER SUPPLYING FILM COOLING TO DOWNSTREAM STATIONS.	NFLOEST 0432
C		NFLOEST 0433
100	CONTINUE	NFLOEST 0434
C	INITIALIZE HOLE CHOKING INDICATOR TO UNCHOKED	NFLOEST 0435
	DO 101 I = 1, NSTA	NFLOEST 0436
101	WCHK(I) = UNCHKD	NFLOEST 0437
	ICHOKE = 0	NFLOEST 0438
	N = NSTA-1	NFLOEST 0439
C		NFLOEST 0440
C	N = NODE NUMBER OF LAST FLOW CHANNEL NODE, AT EXIT OF TRAILING EDGE	NFLOEST 0441
C		NFLOEST 0442
C		NFLOEST 0443
C	CALCULATE IMPINGEMENT FLOWS, AND FORWARD REGION FILM COOLING FLOWS	NFLOEST 0444
C		NFLOEST 0445
	TMP = TOG	NFLOEST 0446
	CALL GASTBL (TMP, C, CP, GAM, PD, R, XMU)	NFLOEST 0447
	GAMO = GAM	NFLOEST 0448
	CPO = CP	NFLOEST 0449
	NODSF = 5*NFWD	NFLOEST 0450
C		NFLOEST 0451
	PAVG = P(2, ISLICE, 1)	NFLOEST 0452
	WJ(ISLICE, 1) = 0.0	NFLOEST 0453
	IF (PAVG.GT.PIM) GO TO 120	NFLOEST 0454
	WCR=CD*PAVG*AJET(1)/(R*TOG)*SQRT(64.4*GAMO*R*TOG/(GAMO+1.))*	NFLOEST 0455
Z	(PIM/PAVG)**((GAMO-1.0)/GAMO)	NFLOEST 0456
	WJ(ISLICE, 1)=PAVG/(R*TOG)*AJET(1)*CD*	NFLOEST 0457
Z	SQRT(64.4*GAMO*R*TOG/(GAMO-1.))* (1.-(PAVG/PIM)**((GAMO-1.)/GAMO))	NFLOEST 0458
1	* (PIM/PAVG)**((GAMO-1.0)/GAMO)	NFLOEST 0459
	WFC(1) = 0.0	NFLOEST 0460
	IF (P(2, ISLICE, 1).GT.PG(1)) WFC(1) = CD*.25*3.1415926*(DHF(1)**2)	NFLOEST 0461
Z	*SQRT(32.2*P(2, ISLICE, 1)*(P(2, ISLICE, 1)-PG(1))/(R*T(2, ISLICE, 5)))	NFLOEST 0462
	IF(WCR.LT.WJ(ISLICE, 1)) WCHK(1) = CHKD	NFLOEST 0463
	IF(WCR.LT.WJ(ISLICE, 1)) WJ(ISLICE, 1)=WCR	NFLOEST 0464
C		NFLOEST 0465
120	PAVG = P(2, ISLICE, 2)	NFLOEST 0466
	WJ(ISLICE, 2) = 0.0	NFLOEST 0467
	IF (PAVG.GT.PIM) GO TO 130	NFLOEST 0468
	WCR=CD*PAVG*AJET(2)/(R*TOG)*SQRT(64.4*GAMO*R*TOG/(GAMO+1.))*	NFLOEST 0469
Z	(PIM/PAVG)**((GAMO-1.0)/GAMO)	NFLOEST 0470
	WJ(ISLICE, 2)=PAVG/(R*TOG)*AJET(2)*CD*	NFLOEST 0471
Z	SQRT(64.4*GAMO*R*TOG/(GAMO-1.))* (1.-(PAVG/PIM)**((GAMO-1.)/GAMO))	NFLOEST 0472
Z	* (PIM/PAVG)**((GAMO-1.0)/GAMO)	NFLOEST 0473
	WFC(2) = 0.0	NFLOEST 0474
	IF (P(2, ISLICE, 2).GT.PG(2)) WFC(2) = CD*.25*3.1415926*(DHF(2)**2)	NFLOEST 0475
Z	*SQRT(32.2*P(2, ISLICE, 2)*(P(2, ISLICE, 2)-PG(2))/(R*T(2, ISLICE, 10)))	NFLOEST 0476
	IF(WCR.LT.WJ(ISLICE, 2)) WCHK(2) = CHKD	NFLOEST 0477
	IF(WCR.LT.WJ(ISLICE, 2)) WJ(ISLICE, 2)=WCR	NFLOEST 0478
C		NFLOEST 0479
		NFLOEST 0480

130	PAVG = P(2, ISLICE, 3)	NFLOEST 0481
	WJ(ISLICE, 3) = 0.0	NFLOEST 0482
	IF (PAVG.GT.PIM) GO TO 140	NFLOEST 0483
	WCR=CD*PAVG*AJET(3)/(R*T0G)*SQRT(64.4*GAMO*R*T0G/(GAMO+1.))*	NFLOEST 0484
Z	(PIM/PAVG)**((GAMO-1.0)/GAMO)	NFLOEST 0485
	WJ(ISLICE, 3)=PAVG/(R*T0G)*AJET(3)*CD*	NFLOEST 0486
Z	SQRT(64.4*GAMO*R*T0G/(GAMO-1.))* (1.-(PAVG/PIM)**((GAMO-1.)/GAMO))	NFLOEST 0487
Z	*(PIM/PAVG)**((GAMO-1.0)/GAMO)	NFLOEST 0488
	WFC(3) = 0.0	NFLOEST 0489
	IF (P(2, ISLICE, 3).GT.PG(3)) WFC(3) = CD*.25*3.1415926*(DHF(3)**2)	NFLOEST 0490
Z	*SQRT(32.2*P(2, ISLICE, 3)*(P(2, ISLICE, 3)-PG(3)) /	NFLOEST 0491
Z	(R*T(2, ISLICE, 15)))	NFLOEST 0492
	IF(WCR.LT.WJ(ISLICE, 3)) WCHK(3) = CHKD	NFLOEST 0493
	IF(WCR.LT.WJ(ISLICE, 3)) WJ(ISLICE, 3)=WCR	NFLOEST 0494
140	CONTINUE	NFLOEST 0495
C		NFLOEST 0496
	DO 150 I = 4, NFWD	NFLOEST 0497
	PAVG = P(2, ISLICE, I)	NFLOEST 0498
	WJ(ISLICE, I) = 0.0	NFLOEST 0499
	IF (PAVG.GT.PIM) GO TO 150	NFLOEST 0500
	WCR=CD*PAVG*AJET(I)/(R*T0G)*SQRT(64.4*GAMO*R*T0G/(GAMO+1.))*	NFLOEST 0501
Z	(PIM/PAVG)**((GAMO-1.0)/GAMO)	NFLOEST 0502
	WJ(ISLICE, I)=PAVG/(R*T0G)*AJET(I)*CD*	NFLOEST 0503
Z	SQRT(64.4*GAMO*R*T0G/(GAMO-1.))* (1.-(PAVG/PIM)**((GAMO-1.)/GAMO))	NFLOEST 0504
Z	*(PIM/PAVG)**((GAMO-1.0)/GAMO)	NFLOEST 0505
	WFC(I) = 0.0	NFLOEST 0506
	IF (P(2, ISLICE, I).GT.PG(I)) WFC(I) = CD*.25*3.1415926*(DHF(I)**2)	NFLOEST 0507
Z	*SQRT(32.2*P(2, ISLICE, I)*(P(2, ISLICE, I)-PG(I)) /	NFLOEST 0508
Z	(R*T(2, ISLICE, 5*I)))	NFLOEST 0509
	IF(WCR.LT.WJ(ISLICE, I)) WCHK(I) = CHKD	NFLOEST 0510
	IF(WCR.LT.WJ(ISLICE, I)) WJ(ISLICE, I)=WCR	NFLOEST 0511
150	CONTINUE	NFLOEST 0512
C		NFLOEST 0513
C	CALCULATE CROSSFLOW RATE AT EACH STATION	NFLOEST 0514
C		NFLOEST 0515
200	WCROS(2, ISLICE, JS) = 0.0	NFLOEST 0516
	JDIS = JS/2	NFLOEST 0517
C****	JDIS IS THE DISPLACEMENT OF THE FLOW SPLIT STATION FROM STATION 1	NFLOEST 0518
	JSENS = JS - 2*JDIS	NFLOEST 0519
C*****	JSENS = 0, STATION IS ON SUCTION SIDE (EVEN);	NFLOEST 0520
C	= 1, STATION IS ON PRESSURE SIDE (ODD)	NFLOEST 0521
	JP = JS + 2	NFLOEST 0522
	JM = JS - 2	NFLOEST 0523
	IF (JM.LT.1) JM = 1	NFLOEST 0524
	JFIN = JS + 1	NFLOEST 0525
	JSTART = JS + 3	NFLOEST 0526
	IF (JS.EQ.1) GO TO 220	NFLOEST 0527
	IF (JSENS.EQ.1) GO TO 230	NFLOEST 0528
	IF (JSENS.EQ.0) GO TO 270	NFLOEST 0529
C		NFLOEST 0530
220	CONTINUE	NFLOEST 0531
C*****	WCROS AT A GIVEN STATION IS THE CROSSFLOW AT UPSTREAM STATION	NFLOEST 0532
C	PLUS IMPINGEMENT JET	NFLOEST 0533
C	FLOW FROM UPSTREAM STATION MINUS ANY FILM COOLING	NFLOEST 0534
C	FLOW AT THIS STATION.	NFLOEST 0535
C		NFLOEST 0536
C**	THIS BLOCK IS EXECUTED IF FLOW SPLIT OCCURS AT STATION 1. ISTART=5	NFLOEST 0537
	WCROS(2, ISLICE, 2) = DELTAN(ISLICE)*(WJ(ISLICE, 1)-WFC(1)) - WFC(2)	NFLOEST 0538
	WCROS(2, ISLICE, 3)=(1.-DELTAN(ISLICE))*(WJ(ISLICE, 1)-WFC(1))-WFC(3)	NFLOEST 0539
	GO TO 320	NFLOEST 0540

C		NFLOEST 0541
230	CONTINUE	NFLOEST 0542
C*****	THIS BLOCK IS EXECUTED IF FLOW SPLIT STATION IS ON	NFLOEST 0543
C	THE PRESSURE SIDE. JSENS=1	NFLOEST 0544
	WCROS(2,ISLICE,JP)=(1.-DELTAN(ISLICE))*(WJ(ISLICE,JS)-WPC(JS))	NFLOEST 0545
	Z-WPC(JP)	NFLOEST 0546
	WCROS(2,ISLICE,JM)=DELTAN(ISLICE)*(WJ(ISLICE,JS)-WPC(JS))-WPC(JM)	NFLOEST 0547
	IF (JM.EQ.1) GO TO 250	NFLOEST 0548
	IRNG = JDIS - 1	NFLOEST 0549
	DO 240 I = 1,IRNG	NFLOEST 0550
	IBK = (JM - 2*I)	NFLOEST 0551
	IBKP = IBK + 2	NFLOEST 0552
240	WCROS(2,ISLICE,IBK) = WCROS(2,ISLICE,IBKP) + WJ(ISLICE,IBKP)	NFLOEST 0553
	Z - WPC(IBK)	NFLOEST 0554
250	JNOD = JFIN	NFLOEST 0555
	DO 260 I = 2,JNOD,2	NFLOEST 0556
	IM8 = I-2	NFLOEST 0557
	IF (IM8.EQ.0) IM8 = 1	NFLOEST 0558
260	WCROS(2,ISLICE,I) = WCROS(2,ISLICE,IM8) + WJ(ISLICE,IM8)-WPC(I)	NFLOEST 0559
	GO TO 320	NFLOEST 0560
C		NFLOEST 0561
270	CONTINUE	NFLOEST 0562
C*****	THIS BLOCK IS EXECUTED IF FLOW SPLIT STATION IS ON	NFLOEST 0563
C	SUCTION SIDE. JSENS=0	NFLOEST 0564
	WCROS(2,ISLICE,JP)=DELTAN(ISLICE)*(WJ(ISLICE,JS)-WPC(JS))-WPC(JP)	NFLOEST 0565
	IF (JS.EQ.2) JM = 1	NFLOEST 0566
	WCROS(2,ISLICE,JM)=(1.-DELTAN(ISLICE))*(WJ(ISLICE,JS)-WPC(JS))	NFLOEST 0567
	Z-WPC(JM)	NFLOEST 0568
	IF (JM.EQ.1) GO TO 300	NFLOEST 0569
	IF (JM.EQ.2) GO TO 290	NFLOEST 0570
	JM1 = JM-2	NFLOEST 0571
	DO 280 I = 2,JM1,2	NFLOEST 0572
	IBK = (JM-I)	NFLOEST 0573
	IBKP = IBK + 2	NFLOEST 0574
280	WCROS(2,ISLICE,IBK) = WCROS(2,ISLICE,IBKP) + WJ(ISLICE,IBKP)	NFLOEST 0575
	Z - WPC(IBK)	NFLOEST 0576
290	WCROS(2,ISLICE,1) = WCROS(2,ISLICE,2) + WJ(ISLICE,2) - WPC(1)	NFLOEST 0577
300	CONTINUE	NFLOEST 0578
C-----	NOW UP THE PRESSURE SIDE	NFLOEST 0579
	JNOD = JFIN	NFLOEST 0580
	DO 310 I = 3,JNOD,2	NFLOEST 0581
310	WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) + WJ(ISLICE,I-2)-WPC(I)	NFLOEST 0582
C		NFLOEST 0583
320	CONTINUE	NFLOEST 0584
	ISTART = JSTART	NFLOEST 0585
	DO 330 I = ISTART,NPFD	NFLOEST 0586
	WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) + WJ(ISLICE,I-2) - WPC(I)	NFLOEST 0587
330	CONTINUE	NFLOEST 0588
C		NFLOEST 0589
C	CALCULATE CROSSFLOW RE & NACH NUMBER SQUARED, AND FILM COOLING RE,	NFLOEST 0590
C	FOR THE FORWARD REGION.	NFLOEST 0591
C		NFLOEST 0592
	DO 360 I = 1,NPFD	NFLOEST 0593
	LCOOL = 5*I	NFLOEST 0594
	LIN = LCOOL-1	NFLOEST 0595
	AHIN = A(LCOOL)	NFLOEST 0596
C		NFLOEST 0597
C	EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL	NFLOEST 0598
C	AND COOLANT BULK	NFLOEST 0599
C		NFLOEST 0600

THP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.	NFLOEST 0601
CALL GASTBL (THP,C,CP,GAM,PD,R,IMU)	NFLOEST 0602
XMUC(I) = XMU	NFLOEST 0603
CPC(I) = CP	NFLOEST 0604
GAMC(I) = GAM	NFLOEST 0605
C	NFLOEST 0606
RE(I) = 12.*3600.*ABS(WCROS(2,ISLICE,I))*DH(I)/(AMIN*XMU)	NFLOEST 0607
REPC(I) = 12.*3600.*WFC(I)/(S(ISLICE)*XMU)	NFLOEST 0608
IF (IHC(I).EQ.3) AMIN = A(LCOOL)*(SP(I)-DP(I))/SP(I)	NFLOEST 0609
W = WCROS(2,ISLICE,I)	NFLOEST 0610
PBAR = P(2,ISLICE,I)	NFLOEST 0611
TBAR = T(2,ISLICE,LCOOL)	NFLOEST 0612
335 AM2(I) = (W/(PBAR*AMIN))**2*R*TBAR/GAMC(I)/32.2	NFLOEST 0613
IF (AM2(I).LT.1.0) GO TO 340	NFLOEST 0614
ANCHOK = AM2(I)	NFLOEST 0615
ICHOK = I	NFLOEST 0616
340 IF (IHC(I).NE.3) GO TO 350	NFLOEST 0617
AM2(I) = AM2(I)*((AMIN/A(LCOOL))**2)	NFLOEST 0618
350 CONTINUE	NFLOEST 0619
360 CONTINUE	NFLOEST 0620
C	NFLOEST 0621
C CALCULATE FLOW DUMPED DIRECTLY INTO TRAILING EDGE REGION.	NFLOEST 0622
C	NFLOEST 0623
PAVG = .5*(P(2,ISLICE,NFWD) + P(2,ISLICE,NFWD-1))	NFLOEST 0624
IF (PAVG.GT.PIN) GO TO 370	NFLOEST 0625
WCR=CD*PAVG*ADUMP/(R*TOG)*SQRT(64.4*GAMO*R*TOG/(GAMO+1.))*	NFLOEST 0626
Z (PIN/PAVG)**((GAMO-1.0)/GAMO)	NFLOEST 0627
WDUMP=PAVG/(R*TOG)*ADUMP*CD*	NFLOEST 0628
Z SQRT(64.4*GAMO*R*TOG/(GAMO-1.))* (1.-(PAVG/PIN)**((GAMO-1.0)/GAMO))	NFLOEST 0629
Z *(PIN/PAVG)**((GAMO-1.0)/GAMO)	NFLOEST 0630
IF(WCR.LT.WDUMP) WCHKDM = CHKD	NFLOEST 0631
IF(WCR.LT.WDUMP) WDUMP=WCR	NFLOEST 0632
C	NFLOEST 0633
C ADD UP TOTAL FLOW FROM IMPINGEMENT PLENUM, WIN.	NFLOEST 0634
C	NFLOEST 0635
370 WIN = WDUMP	NFLOEST 0636
DO 380 I = 1,NSTA	NFLOEST 0637
380 WIN = WIN + WJ(ISLICE,I)	NFLOEST 0638
C	NFLOEST 0639
C TRAILING EDGE REGION, CALCULATE FILM COOLING FLOWS.	NFLOEST 0640
C	NFLOEST 0641
ISTR = NFWD+1	NFLOEST 0642
DO 400 I = ISTR,N,2	NFLOEST 0643
LCOOL = 5*I	NFLOEST 0644
WPCDUM = 0.0	NFLOEST 0645
390 IF (P(2,ISLICE,I).GT.PG(I)) WPCDUM = CD*.25*3.1415926*	NFLOEST 0646
Z SQRT(32.2*P(2,ISLICE,I)*(P(2,ISLICE,I)-PG(I))/	NFLOEST 0647
Z (R*T(2,ISLICE,LCOOL)))	NFLOEST 0648
WFC(I) = WPCDUM*(DHP(I)**2)	NFLOEST 0649
WFC(I+1) = WPCDUM*(DHP(I+1)**2)	NFLOEST 0650
400 CONTINUE	NFLOEST 0651
C	NFLOEST 0652
C TRAILING EDGE REGION, CALCULATE CROSSFLOW, RE, MACH NUMBER	NFLOEST 0653
C SQUARED, AND FILM COOLING RE.	NFLOEST 0654
C	NFLOEST 0655
WCROS(2,ISLICE,ISTR) = WCROS(2,ISLICE,NFWD-1) +	NFLOEST 0656
Z WCROS(2,ISLICE,NFWD) + WDUMP + WJ(ISLICE,NFWD-1)	NFLOEST 0657
Z + WJ(ISLICE,NFWD) - WFC(ISTR) - WFC(ISTR+1)	NFLOEST 0658
WCROS(2,ISLICE,ISTR+1) = WCROS(2,ISLICE,ISTR)	NFLOEST 0659
AMIN = A(NODSP+5)	NFLOEST 0660

C		NFLOEST 0661
C	EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL	NFLOEST 0662
C	AND COOLANT BULK	NFLOEST 0663
C		NFLOEST 0664
	THP = (T(2,ISLICE,5*ISTRT) + T(2,ISLICE,5*ISTRT-1))/2.	NFLOEST 0665
	CALL GASTBL (THP,C,CP,GAM,PD,R,XMU)	NFLOEST 0666
	XHUC(ISTRT) = XMU	NFLOEST 0667
	XHUC(ISTRT+1) = XMU	NFLOEST 0668
	CPC(ISTRT) = CP	NFLOEST 0669
	CPC(ISTRT+1) = CP	NFLOEST 0670
	GAMC(ISTRT) = GAM	NFLOEST 0671
	GAMC(ISTRT+1) = GAM	NFLOEST 0672
C		NFLOEST 0673
	REPC(ISTRT) = 12.*3600.*WFC(ISTRT)/(S(ISLICE)*XMU)	NFLOEST 0674
	REPC(ISTRT+1) = 12.*3600.*WFC(ISTRT+1)/(S(ISLICE)*XMU)	NFLOEST 0675
	RE(ISTRT) = 12.*3600.*ABS(WCROS(2,ISLICE,ISTRT))*DH(ISTRT)/	NFLOEST 0676
Z	(AMIN*XMU)	NFLOEST 0677
	RE(ISTRT+1) = RE(ISTRT)	NFLOEST 0678
	IS = ISTRT	NFLOEST 0679
	IF (IHC(IS).EQ.3) AMIN = A(NODSP+5)*(SP(IS)-DP(IS))/SP(IS)	NFLOEST 0680
	W = WCROS(2,ISLICE,ISTRT)	NFLOEST 0681
	PBAR = P(2,ISLICE,ISTRT)	NFLOEST 0682
	TBAR = T(2,ISLICE,NODSP+5)	NFLOEST 0683
405	AM2(ISTRT) = (W/(PBAR*AMIN))*2*R*TBAR/GAMC(ISTRT)/32.2	NFLOEST 0684
	IF (AM2(ISTRT).LT.1.0) GO TO 410	NFLOEST 0685
	ANCHOK = AM2(ISTRT)	NFLOEST 0686
	ICHOK = ISTRT	NFLOEST 0687
410	IF (IHC(IS).NE.3) GO TO 420	NFLOEST 0688
	AM2(ISTRT) = AM2(ISTRT)*((AMIN/A(NODSP+5))*2)	NFLOEST 0689
420	AM2(ISTRT+1) = AM2(ISTRT)	NFLOEST 0690
	ISTRT = ISTRT + 2	NFLOEST 0691
	IS = NPWD + 1	NFLOEST 0692
	DO 450 I = ISTRT,N,2	NFLOEST 0693
	LCOOL = 5*I	NFLOEST 0694
	AMIN = A(LCOOL)	NFLOEST 0695
	IS = IS + 2	NFLOEST 0696
	IF (IHC(IS).EQ.3) AMIN = A(LCOOL)*(SP(IS)-DP(IS))/SP(IS)	NFLOEST 0697
	WCROS(2,ISLICE,I) = WCROS(2,ISLICE,I-2) - WFC(I) - WFC(I+1)	NFLOEST 0698
	WCROS(2,ISLICE,I+1) = WCROS(2,ISLICE,I)	NFLOEST 0699
C		NFLOEST 0700
C	EVALUATE COOLANT PROPERTIES AT MEAN TEMPERATURE BETWEEN WALL	NFLOEST 0701
C	AND COOLANT BULK	NFLOEST 0702
C		NFLOEST 0703
	THP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LCOOL-1))/2.	NFLOEST 0704
	CALL GASTBL (THP,C,CP,GAM,PD,R,XMU)	NFLOEST 0705
	XHUC(I) = XMU	NFLOEST 0706
	XHUC(I+1) = XMU	NFLOEST 0707
	CPC(I) = CP	NFLOEST 0708
	CPC(I+1) = CP	NFLOEST 0709
	GAMC(I) = GAM	NFLOEST 0710
	GAMC(I+1) = GAM	NFLOEST 0711
C		NFLOEST 0712
	REPC(I) = 12.*3600.*WFC(I)/(S(ISLICE)*XMU)	NFLOEST 0713
	REPC(I+1) = 12.*3600.*WFC(I+1)/(S(ISLICE)*XMU)	NFLOEST 0714
	RE(I) = 12.*3600.*ABS(WCROS(2,ISLICE,I))*DH(I)/(A(LCOOL)*XMU)	NFLOEST 0715
	RE(I+1) = RE(I)	NFLOEST 0716
	W = WCROS(2,ISLICE,I)	NFLOEST 0717
	PBAR = P(2,ISLICE,I)	NFLOEST 0718
	TBAR = T(2,ISLICE,LCOOL)	NFLOEST 0719
425	AM2(I) = (W/(PBAR*AMIN))*2*R*TBAR/GAMC(I)/32.2	NFLOEST 0720

IF (AM2(I).LT.1.0) GO TO 430	NFLOEST 0721
AMCHOK = AM2(I)	NFLOEST 0722
ICHOKE = I	NFLOEST 0723
430 IF (IHC(IS).NE.3) GO TO 440	NFLOEST 0724
AM2(I) = AM2(I)*((AMIN/A(LCOOL))**2)	NFLOEST 0725
440 AM2(I+1) = AM2(I)	NFLOEST 0726
450 CONTINUE	NFLOEST 0727
C	NFLOEST 0728
C CALCULATE COOLANT CHANNEL FRICTION FACTOR AT EACH STATION.	NFLOEST 0729
C	NFLOEST 0730
DO 560 I = 1,NSTA	NFLOEST 0731
LCOOL = 5*I	NFLOEST 0732
IF (WCROS(2,ISLICE,I).LE.0.0) GO TO 550	NFLOEST 0733
C	NFLOEST 0734
C DETERMINE IF RE IS LAMINAR, TRANSITIONAL, OR TURBULENT	NFLOEST 0735
C AND CALCULATE THE FRICTION FACTOR	NFLOEST 0736
C	NFLOEST 0737
IF (IHC(I).EQ.3) GO TO 540	NFLOEST 0738
IF(RE(I).GT.2300.) GO TO 510	NFLOEST 0739
500 FF(I) = DELTA*RE(I)**EPS	NFLOEST 0740
GO TO 560	NFLOEST 0741
510 IF(RE(I).LT.4000.) GO TO 530	NFLOEST 0742
520 FF(I) = ALPHA*RE(I)**BETA	NFLOEST 0743
GO TO 560	NFLOEST 0744
530 A1=DELTA*2300.**EPS	NFLOEST 0745
A2=ALPHA*4000.**BETA	NFLOEST 0746
FF(I) = (A2*(RE(I)-2300.)+A1*(4000.-RE(I)))/1700.	NFLOEST 0747
GO TO 560	NFLOEST 0748
540 CONTINUE	NFLOEST 0749
C	NFLOEST 0750
C FOR A PIN FIN ARRAY:	NFLOEST 0751
FF(I) = 1.060*(RE(I)**(-.3301))	NFLOEST 0752
GO TO 560	NFLOEST 0753
550 CONTINUE	NFLOEST 0754
FF(I) = 0.0	NFLOEST 0755
560 CONTINUE	NFLOEST 0756
C	NFLOEST 0757
C	NFLOEST 0758
C	NFLOEST 0759
C THE FOLLOWING BLOCK IS USED TO COMPUTE THE FILM COOLING EFFECTIVENESS	NFLOEST 0760
C IF IFILM IS SET = 2	NFLOEST 0761
C	NFLOEST 0762
IF (IFILM.LT.2) GO TO 690	NFLOEST 0763
IFSPLT = 0	NFLOEST 0764
DO 610 I = 1,NSTA	NFLOEST 0765
610 XPC(I) = 0.0	NFLOEST 0766
N = NSTA-1	NFLOEST 0767
C LOCATE FILM COOLING HOLES AND SET UP THE XPC ARRAY	NFLOEST 0768
C	NFLOEST 0769
C---IFSPLT IS AN INDICATOR THAT TELLS WHICH SIDE OF THE BLADE STATION 1	NFLOEST 0770
C--- IS TO BE CONSIDERED A PART OF FOR FILM COOLING PURPOSES.	NFLOEST 0771
C--- = 0 IS THE DEFAULT, AND INDICATES SUCTION SIDE	NFLOEST 0772
C--- = 1 WILL INDICATE PRESSURE SIDE.	NFLOEST 0773
C FIRST, MARCH DOWN THE PRESSURE SIDE SEARCHING FOR FILM COOLING HOLES	NFLOEST 0774
IF (DHF(1).GT.0.0) NPC = IFSPLT	NFLOEST 0775
IF (NPC.EQ.0) NPC = NSTA + 1	NFLOEST 0776
XDUM = 0.0	NFLOEST 0777
DO 615 I = 3,NSTA,2	NFLOEST 0778
NOS = 5*I - 4	NFLOEST 0779
IF (I.GT.NPC) XPC(I) = XDUM + DLX(NOS)	NFLOEST 0780

IF (DHF(I).GT.0.0) GO TO 612	NFLOEST 0781
XDUM = XPC(I)	NFLOEST 0782
GO TO 615	NFLOEST 0783
612 NFC = I	NFLOEST 0784
XDUM = 0.0	NFLOEST 0785
615 CONTINUE	NFLOEST 0786
C	NFLOEST 0787
C-- SUCTION SIDE	NFLOEST 0788
C	NFLOEST 0789
IF (DHF(1).GT.0.0) NFC = 1 - IPSPLT	NFLOEST 0790
IF (NFC.EQ.0) NFC = NSTA + 1	NFLOEST 0791
XDUM = 0.0	NFLOEST 0792
DO 625 I = 2,N,2	NFLOEST 0793
NOS = 5*I - 4	NFLOEST 0794
IF (I.GT.NFC) XPC(I) = XDUM + DLX(NOS)	NFLOEST 0795
IF (DHF(I).GT.0.0) GO TO 622	NFLOEST 0796
XDUM = XPC(I)	NFLOEST 0797
GO TO 625	NFLOEST 0798
622 NFC = I	NFLOEST 0799
XDUM = 0.0	NFLOEST 0800
625 CONTINUE	NFLOEST 0801
C INT. J. HT. & MASS TRANS., V8, 1965, PP 55-65	NFLOEST 0802
C FLNEFF = 3.09*((X/(M*S))*(RE*NUC/MUG)**(-1/4) + 4.1)**(-.8)	NFLOEST 0803
C	NFLOEST 0804
IFCS = 0	NFLOEST 0805
IFCP = 0	NFLOEST 0806
IF (WPC(1).LE.0.0) GO TO 630	NFLOEST 0807
IFCS = 1 - IPSPLT	NFLOEST 0808
IFCP = IPSPLT	NFLOEST 0809
IF (RHOVGA(1).GT.0.0) EMES(1) = 144.*WPC(1)/(RHOVGA(1)*S(ISLICE))	NFLOEST 0810
630 CONTINUE	NFLOEST 0811
DO 650 I=2,NSTA	NFLOEST 0812
ISENS = I - 2*(I/2)	NFLOEST 0813
IF (RHOVGA(I).GT.0.0) EMES(I) = 144.*WPC(I)/(RHOVGA(I)*S(ISLICE))	NFLOEST 0814
FLNEFF(I) = 0.0	NFLOEST 0815
IF (ISENS.EQ.0) GO TO 640	NFLOEST 0816
C PRESSURE SIDE SUPPLY HOLE LOCATIONS	NFLOEST 0817
IF (WPC(I).GT.0.0) IFCP = I	NFLOEST 0818
NFCSUP(I) = IFCP	NFLOEST 0819
GO TO 650	NFLOEST 0820
640 CONTINUE	NFLOEST 0821
C SUCTION SIDE SUPPLY HOLE LOCATIONS	NFLOEST 0822
IF (WPC(I).GT.0.0) IFCS = I	NFLOEST 0823
NFCSUP(I) = IFCS	NFLOEST 0824
650 CONTINUE	NFLOEST 0825
C	NFLOEST 0826
TMP = TG(1)	NFLOEST 0827
CALL GASTBL (TMP,C,CPH,GAM,PD,R,XMUM)	NFLOEST 0828
C	NFLOEST 0829
C FINALLY, CALCULATE THE EFFECTIVENESS	NFLOEST 0830
C	NFLOEST 0831
DO 680 I = 1,NSTA	NFLOEST 0832
IMS = NFCSUP(I)	NFLOEST 0833
IF (XPC(I).EQ.0.0.OR.EMES(IMS).EQ.0.0.OR.REPC(IMS).EQ.0.0)	NFLOEST 0834
Z	NFLOEST 0835
C3 = CPC(IMS)/CPH	NFLOEST 0836
XBAR = (XPC(I)/EMES(IMS))*((REPC(IMS)*XNUC(I)/XMUM)**(-.25))	NFLOEST 0837
ETAPRM = 3.09*(XBAR+4.1)**(-.8)	NFLOEST 0838
FLNEFF(I) = C3*ETAPRM/(1.0 + (C3-1.0)*ETAPRM)	NFLOEST 0839
IF (FLNEFF(I).GT.1.0) FLNEFF(I) = 1.0	NFLOEST 0840

680	CONTINUE	NFLOEST 0841
C		NFLOEST 0842
C		NFLOEST 0843
690	CONTINUE	NFLOEST 0844
	RETURN	NFLOEST 0845
	END	NFLOEST 0846
C----	SOURCE.NFLSPLP	NFLSPLP 0847
	SUBROUTINE FLSPLT(AJET, EPSN, ISLICE, NODSF, IDELT, JS, DELTAN, ICONV)	NFLSPLP 0848
	DIMENSION DELTAN(15), AJET(80), JSOLDS(25)	NFLSPLP 0849
C		NFLSPLP 0850
C-	SOURCE.NFLSPLP---A SUBROUTINE TO SET THE STATION AT WHICH COOLING	NFLSPLP 0851
C	AIR FLOW SPLITS BETWEEN THE SUCTION	NFLSPLP 0852
C	AND THE PRESSURE SIDE FLOW CHANNELS.	NFLSPLP 0853
C		NFLSPLP 0854
C	INPUT TO FLSPLT IS THE PRESSURE MATCH PARAMETER, EPSN; THE NO. OF	NFLSPLP 0855
C	NODES (NODSF) IN THE IMPINGEMENT REGION;	NFLSPLP 0856
C	JS COMES IN AS THE CURRENT FLOW SPLIT STATION NO., AND IS	NFLSPLP 0857
C	RETURNED AS THE NEW STATION IF A CHANGE IS NEEDED.	NFLSPLP 0858
C	DELTAN COMES IN AS THE CURRENT FRACTION OF FLOW SPLIT TO	NFLSPLP 0859
C	SUCTION SIDE FROM AN IMPINGEMENT	NFLSPLP 0860
C	JET AT JS. IF A CHANGE IN JS IS NOT NEEDED, DELTAN IS	NFLSPLP 0861
C	USED TO FINE TUNE THE SPLIT.	NFLSPLP 0862
C	ICONV INDICATES IF CONVERGENCE IS COMPLETE.	NFLSPLP 0863
C	= 0--NOT DONE; = 1--OK.	NFLSPLP 0864
C		NFLSPLP 0865
	NFWD = NODSF/5	NFLSPLP 0866
	IF (IUNSTB.EQ.1) GO TO 280	NFLSPLP 0867
	IF (IDELT.NE.1) GO TO 220	NFLSPLP 0868
	JNUMS = 0	NFLSPLP 0869
	IUNSTB = 0	NFLSPLP 0870
	NUMS = 0	NFLSPLP 0871
	JSGNCH=0	NFLSPLP 0872
	JOUTRG=0	NFLSPLP 0873
	DO 210 I = 1,25	NFLSPLP 0874
210	JSOLDS(I) = 0	NFLSPLP 0875
220	CONTINUE	NFLSPLP 0876
	CRITR = .002	NFLSPLP 0877
	ICONV = 0	NFLSPLP 0878
	JSENS = JS - 2*(JS/2)	NFLSPLP 0879
C*****	(SUCTION - PRESSURE SIDE PRESSURES)/ SUCTION SIDE = EPSN	NFLSPLP 0880
	IF (ABS(EPSN).LT.CRITR) GO TO 280	NFLSPLP 0881
C		NFLSPLP 0882
C*****	IF EPSN < 0.0; NEED TO INCREASE FLOW TO PRESSURE SIDE	NFLSPLP 0883
C*****	EPSN > 0.0; NEED TO INCREASE FLOW TO SUCTION SIDE	NFLSPLP 0884
C		NFLSPLP 0885
	IF (JTIMES.EQ.0) GO TO 246	NFLSPLP 0886
C		NFLSPLP 0887
C*****	JTIMES = 0, THIS IS FIRST CHECK AT THIS STATION,	NFLSPLP 0888
C	SO ROUGH ADJUST DELTAN;	NFLSPLP 0889
C*****	1, HAVE BEEN HERE BEFORE, SO FINE TUNE DELTAN.	NFLSPLP 0890
C		NFLSPLP 0891
	IF (JSGNCH.GT.0) GO TO 247	NFLSPLP 0892
C		NFLSPLP 0893
C*****	JSGNCH = 0, THERE HAS NOT BEEN A PRIOR SIGN CHANGE IN EPSN;	NFLSPLP 0894
C*****	= 1, THERE HAS BEEN A SIGN CHANGE BEFORE,	NFLSPLP 0895
C	SO STAY AT THIS STATION	NFLSPLP 0896
C		NFLSPLP 0897
C		NFLSPLP 0898
C		NFLSPLP 0899
242	IF (EPSN/EPNS.LT.0.) JSGNCH = 1	NFLSPLP 0900

C		NFLSPLP 0901
247	CONTINUE	NFLSPLP 0902
	IF (EPLAST/EPSN.GE.0) GO TO 243	NFLSPLP 0903
	DELTAO = DELAST	NFLSPLP 0904
	EPSO = EPLAST	NFLSPLP 0905
243	CONTINUE	NFLSPLP 0906
	IF (JSGNCH.EQ.0) GO TO 252	NFLSPLP 0907
	IF (NUMS.GT.0) GO TO 248	NFLSPLP 0908
	EPSMIN = ABS(EPSN)	NFLSPLP 0909
	DLTAOP = DELTAN(ISLICE)	NFLSPLP 0910
248	NUMS = NUMS + 1	NFLSPLP 0911
	IF (ABS(EPSN).GT.EPSMIN) GO TO 249	NFLSPLP 0912
	EPSMIN = ABS(EPSN)	NFLSPLP 0913
	DLTAOP = DELTAN(ISLICE)	NFLSPLP 0914
249	CONTINUE	NFLSPLP 0915
	IF (NUMS.LT.4) GO TO 252	NFLSPLP 0916
	IF (JNUMS.EQ.1) GO TO 250	NFLSPLP 0917
	NUMS = 0	NFLSPLP 0918
	JNUMS = 1	NFLSPLP 0919
	DELTAN(ISLICE) = DELTAO	NFLSPLP 0920
	JTIMES = 0	NFLSPLP 0921
	JSGNCH = 0	NFLSPLP 0922
	JOUTRG = 0	NFLSPLP 0923
	GO TO 290	NFLSPLP 0924
250	CONTINUE	NFLSPLP 0925
	DELAST = DELTAN(ISLICE)	NFLSPLP 0926
	DELTAN(ISLICE) = DLTAOP	NFLSPLP 0927
	IUNSTB = 1	NFLSPLP 0928
	GO TO 290	NFLSPLP 0929
C		NFLSPLP 0930
C		NFLSPLP 0931
246	JTIMES = 1	NFLSPLP 0932
	EPSO = EPSN	NFLSPLP 0933
	DELTAO = DELTAN(ISLICE)	NFLSPLP 0934
	IF (EPSO.GT.0.0) DELTAN(ISLICE) = (1.0+DELTAN(ISLICE))/2.0	NFLSPLP 0935
	IF (EPSO.LT.0.0) DELTAN(ISLICE) = DELTAN(ISLICE)/2.	NFLSPLP 0936
	IF (DELTAN(ISLICE).EQ.DELTAO) DELTAN(ISLICE) = DELTAN(ISLICE) +	NFLSPLP 0937
Z	(.5-DELTAN(ISLICE))/5.	NFLSPLP 0938
	GO TO 290	NFLSPLP 0939
C		NFLSPLP 0940
C		NFLSPLP 0941
252	CONTINUE	NFLSPLP 0942
	TERM = EPSN*(DELTAO-DELTAN(ISLICE))/(EPSO-EPSN)	NFLSPLP 0943
	IF (TERM.EQ.0.) TERM = .05	NFLSPLP 0944
	IF (JSGNCH.GT.0) GO TO 255	NFLSPLP 0945
	DELTAO = DELTAN(ISLICE)	NFLSPLP 0946
	EPSO = EPSN	NFLSPLP 0947
255	CONTINUE	NFLSPLP 0948
	DELAST = DELTAN(ISLICE)	NFLSPLP 0949
	DELTAN(ISLICE) = DELTAN(ISLICE) - TERM	NFLSPLP 0950
	IF (DELTAN(ISLICE).LT.1.0.AND.DELTAN(ISLICE).GT.0.0) GO TO 290	NFLSPLP 0951
	IF (JOUTRG.GT.0) GO TO 258	NFLSPLP 0952
	IF (DELTAN(ISLICE).LT.0.0) DELTAN(ISLICE) = .01	NFLSPLP 0953
	IF (DELTAN(ISLICE).GT.1.0) DELTAN(ISLICE) = .99	NFLSPLP 0954
	JOUTRG = 1	NFLSPLP 0955
	GO TO 290	NFLSPLP 0956
C		NFLSPLP 0957
C		NFLSPLP 0958
258	CONTINUE	NFLSPLP 0959
	JOUTRG = 0	NFLSPLP 0960

C		NFLSPLP 0961
C	JSGNCH = 0	NFLSPLP 0962
	JTIMES = 0	NFLSPLP 0963
	JSOLDS(JS) = 1	NFLSPLP 0964
	IF (DELTAN(ISLICE).LT.1.) GO TO 265	NFLSPLP 0965
C		NFLSPLP 0966
C***	MOVE JS IN PRESSURE DIRECTION	NFLSPLP 0967
	IF (JSSENS.EQ.0) GO TO 262	NFLSPLP 0968
261	JS = JS + 2	NFLSPLP 0969
	IF (AJET(JS).LE.0.) GO TO 261	NFLSPLP 0970
	GO TO 285	NFLSPLP 0971
C		NFLSPLP 0972
C		NFLSPLP 0973
262	CONTINUE	NFLSPLP 0974
	IF (JS.EQ.2) JS = 1	NFLSPLP 0975
	IF (JS.GT.2) JS = JS - 2	NFLSPLP 0976
	IF (AJET(JS).LE.0.) GO TO 262	NFLSPLP 0977
	GO TO 285	NFLSPLP 0978
C		NFLSPLP 0979
C		NFLSPLP 0980
265	CONTINUE	NFLSPLP 0981
C***	MOVE JS IN SUCTION DIRECTION	NFLSPLP 0982
	IF (JSSENS.EQ.0) GO TO 267	NFLSPLP 0983
	IF (JS.EQ.1) JS = 2	NFLSPLP 0984
	IF (JS.GE.3) JS = JS - 2	NFLSPLP 0985
	IF (AJET(JS).GT.0.) GO TO 265	NFLSPLP 0986
	JSSENS = JS - 2*(JS/2)	NFLSPLP 0987
	GO TO 265	NFLSPLP 0988
C		NFLSPLP 0989
C		NFLSPLP 0990
267	CONTINUE	NFLSPLP 0991
	JS = JS + 2	NFLSPLP 0992
	IF (AJET(JS).LE.0.) GO TO 267	NFLSPLP 0993
	GO TO 285	NFLSPLP 0994
C		NFLSPLP 0995
C*****	GET READY TO LEAVE SUBROUTINE	NFLSPLP 0996
C		NFLSPLP 0997
C	THIS BLOCK IS EXECUTED IF CONVERGENCE WAS DETECTED	NFLSPLP 0998
C		NFLSPLP 0999
280	ICONV = 1	NFLSPLP 1000
	JTIMES = 0	NFLSPLP 1001
	IF (IUNSTB.EQ.1) WRITE(6,284) ISLICE,IDELT,JS,DELTAN(ISLICE)	NFLSPLP 1002
	IF (IUNSTB.EQ.1) WRITE(8,284) ISLICE,IDELT,JS,DELTAN(ISLICE)	NFLSPLP 1003
	IUNSTB = 0	NFLSPLP 1004
	EPLAST = EPSN	NFLSPLP 1005
	RETURN	NFLSPLP 1006
284	FORMAT(1H2,40(' '),40('*'))// ' SLICE ',I2,', POOR FLOW SPLIT, ',	NFLSPLP 1007
Z	I3,' ITERATIONS, SPLIT AT STATION ',	NFLSPLP 1008
Z	I2,', BEST SPLIT IS AT DELTA = ',P6.4)	NFLSPLP 1009
C		NFLSPLP 1010
C		NFLSPLP 1011
C	THIS BLOCK IS EXECUTED FOR AN ABNORMAL EXIT---PROGRAM IS TERMINATED	NFLSPLP 1012
C		NFLSPLP 1013
789	CONTINUE	NFLSPLP 1014
	WRITE(6,792) DELTAN(ISLICE)	NFLSPLP 1015
	WRITE(8,792) DELTAN(ISLICE)	NFLSPLP 1016
	STOP	NFLSPLP 1017
792	FORMAT(/5X,' ***** FLOW SPLIT CANNOT BE MADE AS SPECIFIED,',	NFLSPLP 1018
Z	' DELTA = ',P9.5)	NFLSPLP 1019
		NFLSPLP 1020

C		NFLSPLP 1021
C		NFLSPLP 1022
285	IF (JSOLDS(JS).EQ.1) GO TO 789	NFLSPLP 1023
	DELTAN(ISLICE) = .50	NFLSPLP 1024
C		NFLSPLP 1025
C		NFLSPLP 1026
C	THIS BLOCK IS THE USUAL EXIT AFTER ADJUSTING THE FLOW SPLIT	NFLSPLP 1027
C		NFLSPLP 1028
290	CONTINUE	NFLSPLP 1029
	EPLAST = EPSN	NFLSPLP 1030
	IF (JSGNCH.EQ.0) DELAST = DELTAO	NFLSPLP 1031
	IDELT = IDELT + 1	NFLSPLP 1032
	RETURN	NFLSPLP 1033
	END	NFLSPLP 1034
C----	SOURCE.NGASDAT	NGASDAT 1035
	BLOCK DATA	NGASDAT 1036
C		NGASDAT 1037
C--	SOURCE.NGASDAT---	NGASDAT 1038
C		NGASDAT 1039
	COMMON /GAAS/ GS(200),NG	NGASDAT 1040
	DATA GS/620., 1160., 1700., 2240., 2780., 3320.,	NGASDAT 1041
Z	.02564, .03580, .04548, .05467, .06435, .07475,	NGASDAT 1042
Z	.2511, .2681, .2814, .2939, .3070, .3214,	NGASDAT 1043
Z	.706, .706, .705, .703, .702, .699,	NGASDAT 1044
Z	.07233, .09458, .11369, .13063, .14683, .16256,	NGASDAT 1045
Z	170*0.0/, NG /6/	NGASDAT 1046
C		NGASDAT 1047
C---	GS IS TABLE OF AIR PROPERTIES VS TEMPERATURE AT CONSTANT PRESSURE	NGASDAT 1048
C---	PROPERTY VALUES ARE FROM POFERL & SVEHLA, TN D-7488, AT 20 ATM.	NGASDAT 1049
C---	NG IS THE NUMBER OF TEMPERATURE ENTRIES IN THE TABLE	NGASDAT 1050
C---	ENTRIES IN GS ARE:	NGASDAT 1051
C---	1ST NG ARE TEMPERATURE, (F)	NGASDAT 1052
C---	2ND NG ARE THERMAL CONDUCTIVITY, (BTU/(HR*FT*R))	NGASDAT 1053
C---	3RD NG ARE SPECIFIC HEAT, (BTU/(LBM*R))	NGASDAT 1054
C---	4TH NG ARE PRANDTL NUMBER	NGASDAT 1055
C---	5TH NG ARE VISCOSITY, (LBM/(FT*HR))	NGASDAT 1056
C		NGASDAT 1057
	END	NGASDAT 1058
C----	SOURCE.NGASTB	NGASTB 1059
	SUBROUTINE GASTBL(TMP,C,CP,GAM,PD,R,XMU)	NGASTB 1060
C		NGASTB 1061
C-	SOURCE.NGASTB	NGASTB 1062
C		NGASTB 1063
C	A SUBROUTINE TO LOOK UP GAS PROPERTIES IN AN INPUT TABLE (GS(200))	NGASTB 1064
C	WHERE TMP = TEMPERATURE AT WHICH PROPERTIES ARE TO BE EVALUATED (R)	NGASTB 1065
C	C = GAS THERMAL CONDUCTIVITY (BTU/(HR*FT*R))	NGASTB 1066
C	CP = GAS SPECIFIC HEAT (BTU/(LBM*R))	NGASTB 1067
C	GAM = RATIO OF SPECIFIC HEATS	NGASTB 1068
C	PD = PRANDTL NUMBER	NGASTB 1069
C	R = SPECIFIC GAS CONSTANT (FT*LBF)/(LBM*R)	NGASTB 1070
C	XMU = VISCOSITY (LBM/(FT*HR))	NGASTB 1071
C		NGASTB 1072
	COMMON /GAAS/ GS(200),NG	NGASTB 1073
	DIMENSION AC(5)	NGASTB 1074
C		NGASTB 1075
	TMP1=TMP - 460.	NGASTB 1076
	IF(TMP1.GT.GS(1)) GO TO 200	NGASTB 1077
100	AP1=0.0	NGASTB 1078
	AP2=1.0	NGASTB 1079
	I1=2	NGASTB 1080

	I2=1	NGASTB 1081
	GO TO 500	NGASTB 1082
200	DO 300 I=1,NG	NGASTB 1083
	I1=I	NGASTB 1084
	IF(GS(I).GT.TMP1) GO TO 400	NGASTB 1085
300	CONTINUE	NGASTB 1086
	TMP1=GS(NG)	NGASTB 1087
400	I2=I1-1	NGASTB 1088
	AP1=(TMP1-GS(I2))/(GS(I1)-GS(I2))	NGASTB 1089
	AP2=1.0-AP1	NGASTB 1090
500	DO 600 J=1,4	NGASTB 1091
	I1=I1+NG	NGASTB 1092
	I2=I2+NG	NGASTB 1093
600	AC(J)=AP1*GS(I1)+AP2*GS(I2)	NGASTB 1094
	AC(5)=1.0/(1.0-R/(778.2*AC(2)))	NGASTB 1095
	C=AC(1)	NGASTB 1096
	CP=AC(2)	NGASTB 1097
	PD=AC(3)	NGASTB 1098
	XHU=AC(4)	NGASTB 1099
	GAM=AC(5)	NGASTB 1100
	RETURN	NGASTB 1101
	END	NGASTB 1102

C----	SOURCE.NGAUS	NGAUS 1103
	SUBROUTINE GAUSS(N,K)	NGAUS 1104
C		NGAUS 1105
C-	SOURCE.NGAUS----	NGAUS 1106
C		NGAUS 1107
C----	GIVEN A COMPRESSED VERSION OF AN AUGMENTED, BAND MATRIX A	NGAUS 1108
C----	WHERE K IS THE WIDTH OF THE BAND, N IS THE NUMBER OF ROWS (MAX 400)	NGAUS 1109
C----	DIAGONAL ELEMENTS OF THE ORIGINAL MATRIX ARE STORED IN	NGAUS 1110
C----	COLUMN ((K/2)+1) -- TCOP(I,((K/2)+1))	NGAUS 1111
C----	THE ORIGINAL RIGHT HAND SIDE IS IN COLUMN K+1--- TCOP(I,K+1)	NGAUS 1112
C----	GAUSS ELIMINATION IS USED TO MAKE ALL ELEMENTS BELOW	NGAUS 1113
C----	THE DIAGONAL ZERO.	NGAUS 1114
C----	BACK-SUBSTITUTION IS USED TO COMPUTE THE X'S, WHICH ARE	NGAUS 1115
C----	RETURNED IN TCOP(I,K+1)	NGAUS 1116
C		NGAUS 1117
	REAL*8 TCOP	NGAUS 1118
	COMMON /MATRIX/ TCOP(400,30)	NGAUS 1119
	IWR = 0	NGAUS 1120
	NROW = 0	NGAUS 1121
	IF (IWR.EQ.0) GO TO 63	NGAUS 1122
C		NGAUS 1123
C	DEBUGGING OUTPUT:	NGAUS 1124
C		NGAUS 1125
C	IWR CAN BE SET DYNAMICALLY IN ORDER TO GET DEBUG OUTPUT OF	NGAUS 1126
C	SELECTED ROWS OF THE MATRIX, BEFORE OR AFTER REDUCTION.	NGAUS 1127
C		NGAUS 1128
	WRITE(8,57)	NGAUS 1129
57	FORMAT(' ENTER NUMBER OF ROW TO BE DISPLAYED. USE I3 FORMAT')	NGAUS 1130
58	READ(7,59) NROW	NGAUS 1131
59	FORMAT(I3)	NGAUS 1132
	IF (NROW.EQ.0) GO TO 63	NGAUS 1133
	KP = K+1	NGAUS 1134
	WRITE(8,60) NROW	NGAUS 1135
	WRITE(8,61) (I,TCOP(NROW,I),I=1,KP)	NGAUS 1136
60	FORMAT(/' TCOP MATRIX, ROW NO. ',I3)	NGAUS 1137
61	FORMAT(5(' ',I3,',',',D17.10,')'))	NGAUS 1138
	WRITE(8,62)	NGAUS 1139
62	FORMAT(/' ENTER ANOTHER ROW NO. OR 000 TO CONTINUE PROCESSING')	NGAUS 1140


```

      GO TO 58
C
C
63  CONTINUE
65  JPIV = K/2 + 1
      N1 = N-1
      DO 100 I = 1,N1
      JS = I+1
      JF = I + K/2
      IF (JF.GT.N) JF = N
      PIVOT = TCOF(I,JPIV)
      IF (PIVOT.EQ.0.0) GO TO 130
      DO 90 J = JS,JF
      JR = JPIV-J+I
      IF (DABS(TCOF(J,JR)).LT.1.0D-30) GO TO 90
      FM = TCOF(J,JR)/PIVOT
      TCOF(J,JR) = 0.0
      LS = JR + 1
      LF = LS + K/2
      IF (LF.LT.LS) GO TO 85
      DO 80 L = LS,LF
      LR = L+JPIV+1-LS
      IF (LR.GT.K) GO TO 85
80   TCOF(J,L) = TCOF(J,L) - FM*TCOF(I,LR)
85   TCOF(J,K+1) = TCOF(J,K+1) - FM*TCOF(I,K+1)
90   CONTINUE
100  CONTINUE
C
C
155  IF (IWR.EQ.0) GO TO 163
C
C  DEBUGGING OUTPUT:
C
      WRITE(8,57)
158  READ(7,59) NROW
      IF (NROW.EQ.0) GO TO 163
      KP = K+1
      WRITE(8,60) NROW
      WRITE(8,61) (I,TCOF(NROW,I),I=1,KP)
      WRITE(8,62)
      GO TO 158
C
C
163  CONTINUE
      TCOF(N,K+1) = TCOF(N,K+1)/TCOF(N,JPIV)
      DO 120 I = 1,N1
      IIN = N-I
      JF = K/2
      SUM = TCOF(IIN,K+1)
      DO 115 J = 1,JF
      JP = J + JPIV
      IJ = J+IIN
      IF (IJ.GT.N) GO TO 117
115  SUM = SUM - TCOF(IJ,K+1)*TCOF(IIN,JP)
117  CONTINUE
120  TCOF(IIN,K+1) = SUM/TCOF(IIN,JPIV)
125  CONTINUE
      RETURN
130  WRITE(7,135) I
135  FORMAT(/'  DIAGONAL ELEMENT FOR ROW ',I2,' IS ZERO. NO ',

```

```

NGAUS 1141
NGAUS 1142
NGAUS 1143
NGAUS 1144
NGAUS 1145
NGAUS 1146
NGAUS 1147
NGAUS 1148
NGAUS 1149
NGAUS 1150
NGAUS 1151
NGAUS 1152
NGAUS 1153
NGAUS 1154
NGAUS 1155
NGAUS 1156
NGAUS 1157
NGAUS 1158
NGAUS 1159
NGAUS 1160
NGAUS 1161
NGAUS 1162
NGAUS 1163
NGAUS 1164
NGAUS 1165
NGAUS 1166
NGAUS 1167
NGAUS 1168
NGAUS 1169
NGAUS 1170
NGAUS 1171
NGAUS 1172
NGAUS 1173
NGAUS 1174
NGAUS 1175
NGAUS 1176
NGAUS 1177
NGAUS 1178
NGAUS 1179
NGAUS 1180
NGAUS 1181
NGAUS 1182
NGAUS 1183
NGAUS 1184
NGAUS 1185
NGAUS 1186
NGAUS 1187
NGAUS 1188
NGAUS 1189
NGAUS 1190
NGAUS 1191
NGAUS 1192
NGAUS 1193
NGAUS 1194
NGAUS 1195
NGAUS 1196
NGAUS 1197
NGAUS 1198
NGAUS 1199
NGAUS 1200

```

Z	'FURTHER ATTEMPT TO SOLVE WILL BE MADE.'	NGAUS	1201
	GO TO 125	NGAUS	1202
	END	NGAUS	1203

C----	SOURCE.NGETINT		NGETINT	1204
	SUBROUTINE GETIN		NGETINT	1205
C			NGETINT	1206
C-	SOURCE.NGETINT----		NGETINT	1207
C			NGETINT	1208
	COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000),		NGETINT	1209
Z	BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),		NGETINT	1210
Z	BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),		NGETINT	1211
Z	QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80),		NGETINT	1212
Z	QTIPIN(400), QTIP(80), RHOVG(400), PEX(400),		NGETINT	1213
Z	BCTIME(50), TTIO(50), PTIO(50), WPLEN,		NGETINT	1214
Z	WSVST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCP		NGETINT	1215
C			NGETINT	1216
	COMMON /FLMCO/ RHOVG(80), PG(80), XPC(80), FLMEFF(80),		NGETINT	1217
Z	XNUC(80), EMES(80), REFC(80), NFCSUP(80)		NGETINT	1218
C			NGETINT	1219
	COMMON /IMPCOR/ CIMP1, CIMP2, CIMP3, CIMP4, CIMP5, CIMP6, CIMP7,		NGETINT	1220
Z	DIMP1, DIMP2, DIMP3, DIMP4, DIMP5, DIMP6		NGETINT	1221
C			NGETINT	1222
	COMMON /RADL/ APLN(15), DPLN(15), RIN(15), ROUT(15),		NGETINT	1223
Z	PIN(15), TIN(15), W(15), WS		NGETINT	1224
C			NGETINT	1225
	COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000),		NGETINT	1226
Z	IPLOT, MD1, MD2, MD3, IADJIN, IWRITE		NGETINT	1227
C			NGETINT	1228
	COMMON /TCO/ ADUMP, BTA, CD, CP,		NGETINT	1229
Z	GAM, PIM, R, SPAN, TOG,		NGETINT	1230
Z	WDUMP, WIM, AKC(15,80), AKW(15,80),		NGETINT	1231
Z	A(400), AJET(80), AM2(80), CNUM(80),		NGETINT	1232
Z	DH(80), DHF(80), DHJ(80),		NGETINT	1233
Z	DLX(400), FF(80), HC(80), HG(80),		NGETINT	1234
Z	P(2,15,80), PEXIT(15), PUMP(80), QG(80),		NGETINT	1235
Z	QSNK(80), RR(80), S(15), T(2,15,400),		NGETINT	1236
Z	TG(80), TAU(400), WFC(80),		NGETINT	1237
Z	WJ(15,80), WCROS(2,15,80), IN(80),		NGETINT	1238
Z	ICOR, IFILM, IHUB, ITIP,		NGETINT	1239
Z	ISBLOK, ISLICE, NBLKSZ, NSLICE,		NGETINT	1240
Z	NFWD, NSTA, IHC(80)		NGETINT	1241
C			NGETINT	1242
	COMMON /TRNSNT/ RHOC, RHOM, SPHTC, SPHTM,		NGETINT	1243
Z	DLTYME, TYME, TEPS, TYMMAX		NGETINT	1244
C			NGETINT	1245
	COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSFL(2),		NGETINT	1246
Z	CTHPF(2), CTCON(2), CDEN(2), CSPHT(2), CGASC(2),		NGETINT	1247
Z	CVISC(2), CRHOVG(2), IUNITS		NGETINT	1248
C			NGETINT	1249
	DIMENSION THK(3), TDLX(5), TFLMHL(10)		NGETINT	1250
C			NGETINT	1251
	NAMelist /TITL/ TITLE		NGETINT	1252
C			NGETINT	1253
	NAMelist /CHANLS/ NSLICE, NSTA, INEDIT, IPLOT, IWRITE,		NGETINT	1254
Z	MD1, MD2, MD3, IUNITS, IFILM, IADJIN		NGETINT	1255
C			NGETINT	1256
	NAMelist /BC/ N9CS, NBCP, BCXS, BCXP, BCHGS, BCHGP,		NGETINT	1257
Z	BCTGS, BCTGP, BCQGS, BCQGP, BCPGS, BCPGP,		NGETINT	1258
Z	THUBIN, QHUBIN, TTIPIN, QTIPIN, RHOVG,		NGETINT	1259
Z	PEX, BCTIME, TTIO, PTIO, WPLEN,		NGETINT	1260

Z	WSVST,	AKCTBL,	AKWTBL,	RHOC,	RHOM,	NGETINT	1261
Z	SPHTC,	SPHTM,	DLTYME,	TEPS,	TYMMAX	NGETINT	1262
C	NAMELIST /CONTRL/ NFWD, ICOR, NCEO					NGETINT	1263
C	NAMELIST /PROPS/ CD, SPAN, ADUMP, DHYD, APLEN, RO,					NGETINT	1264
Z	RI,	CIMP1,	CIMP2,	CIMP3,	CIMP4,	NGETINT	1265
Z	CIMP6,	CIMP7,	DIMP1,	DIMP2,	DIMP3,	NGETINT	1266
Z	DIMP5,	DIMP6				NGETINT	1267
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1268
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1269
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1270
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1271
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1272
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1273
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1274
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1275
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1276
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1277
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1278
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1279
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1280
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1281
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1282
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1283
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1284
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1285
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1286
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1287
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1288
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1289
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1290
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1291
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1292
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1293
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1294
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1295
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1296
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1297
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1298
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1299
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1300
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1301
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1302
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1303
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1304
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1305
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1306
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1307
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1308
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1309
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1310
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1311
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1312
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1313
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1314
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1315
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1316
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1317
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1318
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1319
C	NAMELIST /GEO/ ISTA, ISTB, THK, TDLX, TDHJ, TXN,					NGETINT	1320
Z	TDHF,	TRR,	IHCT,	TDP,	TSP,	NGETINT	1320

C	DLTIME = TIME STEP SIZE FOR TRANSIENT CALCULATIONS (SEC)	NGETINT	1321
C	TYMMAX = MAX. TIME (SEC) TO WHICH TRANSIENT IS CARRIED.	NGETINT	1322
C	TEPS = FRACTION OF TIME STEP AT WHICH TEMP. IS EVALUATED. (NEW = OLD	NGETINT	1323
C	+ TEPS*(NEW-OLD))	NGETINT	1324
C	WSVST = TABLE OF WHEEL SPEED VS TIME, (RPM VS SEC), ODD SUBSCRIPTS	NGETINT	1325
C	ARE SPEED,EVEN ARE TIME, WSVST(2)=0.0	NGETINT	1326
C		NGETINT	1327
C	CIMP1 TO CIMP5 ARE EXPONENTS TO BE USED IN A GENERAL	NGETINT	1328
C	IMPINGEMENT WITH CROSSFLOW CORRELATION. IF NOT SPECIFIED, THEN THEN	NGETINT	1329
C	BUILT IN KIRCHER-TABAKOFF CORRELATION IS USED.	NGETINT	1330
C	SEE SUBROUTINE HCOOLT FOR DESCRIPTION OF GENERAL CORRELATION.	NGETINT	1331
C		NGETINT	1332
C	INITIALIZE:	NGETINT	1333
C		NGETINT	1334
100	CONTINUE	NGETINT	1335
	IEND = 0	NGETINT	1336
	IADJIN = 0	NGETINT	1337
	IHUB = 2	NGETINT	1338
	ITIP = 2	NGETINT	1339
	CIMP1 = 0.0	NGETINT	1340
	DIMP1 = 0.0	NGETINT	1341
	ADUMP = 0.0	NGETINT	1342
	IPFILM = 0	NGETINT	1343
	IUNITS = 2	NGETINT	1344
	ALPHA = .04	NGETINT	1345
	BETA = -.16	NGETINT	1346
	DELTA = 16.	NGETINT	1347
	EPS = -1.	NGETINT	1348
	CD = .8	NGETINT	1349
C		NGETINT	1350
C	GAS CONSTANT FOR AIR, FT LBF/LBM R	NGETINT	1351
	R = 53.35	NGETINT	1352
C		NGETINT	1353
C--	SET VALUES FOR UNITS CORRECTION FACTORS---	NGETINT	1354
C--	... (1) CONVERTS FROM SI TO ENGLISH, ... (2) MAKES NO CONVERSION--	NGETINT	1355
C	ALREADY IN ENGLISH	NGETINT	1356
C		NGETINT	1357
C---	CINCH(1) IS CONVERSION FACTOR FROM (CM) TO (IN)	NGETINT	1358
	CINCH(1) = .39370	NGETINT	1359
	CINCH(2) = 1.0	NGETINT	1360
C---	CHTC(1) IS CONVERSION FACTOR FROM (WATTS/M**2 K) TO (BTU/HR FT**2R)	NGETINT	1361
	CHTC(1) = .17623	NGETINT	1362
	CHTC(2) = 1.0	NGETINT	1363
C---	CHFLX(1) IS CONVERSION FACTOR FROM (WATTS/M**2) TO (BTU/HR FT**2)	NGETINT	1364
	CHFLX(1) = .31721	NGETINT	1365
	CHFLX(2) = 1.0	NGETINT	1366
C---	CPRSP(1) IS CONVERSION FACTOR FROM (KILOPASCALS) TO (PSIA)	NGETINT	1367
	CPRSR(1) = .14503	NGETINT	1368
	CPRSE(2) = 1.0	NGETINT	1369
C---	CMSFL(1) IS CONVERSION FACTOR FROM (KG/HR) TO (LBM/HR)	NGETINT	1370
	CMSPL(1) = 2.67924	NGETINT	1371
	CMSPL(2) = 1.0	NGETINT	1372
C---	CTHPP(1) IS CONVERSION FACTOR FROM (K) TO (R)	NGETINT	1373
	CTHPP(1) = 1.8	NGETINT	1374
	CTHPP(2) = 1.0	NGETINT	1375
C---	CTCON(1) IS CONVERSION FACTOR FROM (WATTS/M K) TO (BTU/HR FT R)	NGETINT	1376
	CTCON(1) = .57817	NGETINT	1377
	CTCON(2) = 1.0	NGETINT	1378
C---	CDEN(1) IS CONVERSION FACTOR FROM (KG/M**3) TO (LBM/FT**3)	NGETINT	1379
	CDEN(1) = .06243	NGETINT	1380

	CDEN(2) = 1.0	NGETINT 1381
C---	CSPHT(1) IS CONVERSION FACTOR FROM (J/KG K) TO (BTU/LBM R)	NGETINT 1382
	CSPHT(1) = .000239	NGETINT 1383
	CSPHT(2) = 1.0	NGETINT 1384
C---	CVISC(1) IS CONVERSION FACTOR FROM (PA SEC) TO (LBM/FT HR)	NGETINT 1385
	CVISC(1) = 2419.096	NGETINT 1386
	CVISC(2) = 1.0	NGETINT 1387
C---	CGASC(1) IS CONVERSION FROM (J/KG K) TO (FT LBF/LBM R)	NGETINT 1388
	CGASC(1) = .18602	NGETINT 1389
	CGASC(2) = 1.0	NGETINT 1390
C---	CRHOVG IS CONVERSION FROM (KG/SEC M**2) TO (LBM/SEC FT**2)	NGETINT 1391
	CRHOVG(1) = .0204823	NGETINT 1392
	CRHOVG(2) = 1.0	NGETINT 1393
C		NGETINT 1394
	DO 105 I = 1,30	NGETINT 1395
105	TITLE(I) = TIKLE	NGETINT 1396
C		NGETINT 1397
	DO 106 I = 1,1000	NGETINT 1398
	BCHGS(I) = 0.0	NGETINT 1399
	BCHGP(I) = 0.0	NGETINT 1400
	BCTGS(I) = 0.0	NGETINT 1401
	BCTGP(I) = 0.0	NGETINT 1402
	BCQGS(I) = 0.0	NGETINT 1403
	BCQGP(I) = 0.0	NGETINT 1404
	BCPGS(I) = 0.0	NGETINT 1405
106	BCPGP(I) = 0.0	NGETINT 1406
	RHOC = 0.0	NGETINT 1407
	RHOM = 0.0	NGETINT 1408
	SPHTC = 0.0	NGETINT 1409
	SPHTM = 0.0	NGETINT 1410
	DO 107 I = 1,400	NGETINT 1411
	THUBIN(I) = 0.0	NGETINT 1412
	QHUBIN(I) = 0.0	NGETINT 1413
	TTIPIN(I) = 0.0	NGETINT 1414
	QTIPIN(I) = 0.0	NGETINT 1415
	RHOVG(I) = 0.0	NGETINT 1416
107	PEX(I) = 0.0	NGETINT 1417
	DO 108 I = 1,50	NGETINT 1418
	BCTIME(I) = 0.0	NGETINT 1419
	TTIO(I) = 0.0	NGETINT 1420
	PTIO(I) = 0.0	NGETINT 1421
	WSVST(I) = 0.0	NGETINT 1422
108	RR(I) = 0.0	NGETINT 1423
C		NGETINT 1424
	DO 110 I = 1,6000	NGETINT 1425
110	CHANL(I) = 0.0	NGETINT 1426
C		NGETINT 1427
	DO 112 I = 1,15	NGETINT 1428
	PEXIT(I) = 0.0	NGETINT 1429
	DO 112 J = 1,80	NGETINT 1430
	AKC(I,J) = 0.0	NGETINT 1431
112	AKW(I,J) = 0.0	NGETINT 1432
C		NGETINT 1433
	DO 115 I = 1,2000	NGETINT 1434
115	INDCHN(I) = 0	NGETINT 1435
	DO 116 I = 1,20	NGETINT 1436
	AKCTBL(I) = 0.0	NGETINT 1437
116	AKWTBL(I) = 0.0	NGETINT 1438
	IPLOT = 0	NGETINT 1439
	IWRITE = 0	NGETINT 1440

INEDIT = 0	NGETINT 1441
TEPS = 1.0	NGETINT 1442
DLTIME = 0.0	NGETINT 1443
READ(5,TITL)	NGETINT 1444
READ(5,CHANLS)	NGETINT 1445
READ(5,BC)	NGETINT 1446
IF (BCHGS(1).EQ.0.0) BTA=1.0	NGETINT 1447
IF (BCQGS(1).EQ.0.0) BTA = 0.0	NGETINT 1448
IF (TTIPIN(1).GT.0.0) ITIP = 1	NGETINT 1449
IF (THUBIN(1).GT.0.0) IHUB = 1	NGETINT 1450
IF (ABS(QTIPIN(1)).GT.0.0) ITIP = 3	NGETINT 1451
IF (ABS(QHUBIN(1)).GT.0.0) IHUB = 3	NGETINT 1452
WS = WSVST(1)	NGETINT 1453
C	NGETINT 1454
PEXIT(1) = PEX(1)	NGETINT 1455
DO 175 ICHLNO = 1,NSLICE	NGETINT 1456
C	NGETINT 1457
C ICHLNO IS THE CHANNEL NUMBER; = 1 AT THE HUB, = NSLICE AT THE TIP	NGETINT 1458
C	NGETINT 1459
READ(5,CONTRL)	NGETINT 1460
NODSF = 5*NFWO	NGETINT 1461
C--NODSF IS THE NUMBER OF NODES IN THE FORWARD REGION	NGETINT 1462
C	NGETINT 1463
NODST = 5*NSTA	NGETINT 1464
C--NODST IS THE TOTAL NUMBER OF NODES IN THE BLADE SLICE ICHLNO	NGETINT 1465
C	NGETINT 1466
NBLKSZ = (15 + 2*NODST) + 8*NSTA	NGETINT 1467
C--NBLKSZ IS THE SIZE OF THE DATA BLOCK RESERVED IN CHANL ARRAY FOR THIS	NGETINT 1468
C SLICE ICHLNO	NGETINT 1469
C	NGETINT 1470
ISBLOK = IEND + 1	NGETINT 1471
C--ISBLOK IS THE STARTING POINT IN CHANL ARRAY FOR THIS BLOCK OF DATA	NGETINT 1472
C	NGETINT 1473
INSTRT = 15 + (ICHLNO-1)*(15 + NSTA)	NGETINT 1474
C--INSTRT IS THE STARTING POINT IN INDCHN ARRAY FOR THIS BLOCK OF	NGETINT 1475
C INTEGER DATA	NGETINT 1476
C	NGETINT 1477
INDCHN(ICHLNO) = INSTRT	NGETINT 1478
INDCHN(INSTRT) = ICHLNO	NGETINT 1479
INDCHN(INSTRT+1) = IFILM	NGETINT 1480
INDCHN(INSTRT+2) = ICOR	NGETINT 1481
INDCHN(INSTRT+3) = NFWO	NGETINT 1482
INDCHN(INSTRT+4) = NSTA	NGETINT 1483
INDCHN(INSTRT+5) = ISBLOK	NGETINT 1484
INDCHN(INSTRT+6) = NBLKSZ	NGETINT 1485
INDCHN(INSTRT+7) = IPLOT	NGETINT 1486
INDCHN(INSTRT+8) = MD1	NGETINT 1487
INDCHN(INSTRT+9) = MD2	NGETINT 1488
INDCHN(INSTRT+10) = MD3	NGETINT 1489
INDCHN(INSTRT+12) = IHUB	NGETINT 1490
INDCHN(INSTRT+13) = ITIP	NGETINT 1491
IIHCTZ = INSTRT + 14	NGETINT 1492
C--IIHCTZ IS THE RELATIVE ZERO POINT IN INDCHN FOR STORAGE OF THE	NGETINT 1493
C INDICATOR IHC	NGETINT 1494
READ(5,PROPS)	NGETINT 1495
S(ICHLNO) = SPAN*CINCH(IUNITS)	NGETINT 1496
APLN(ICHLNO) = APLEN*CINCH(IUNITS)*CINCH(IUNITS)	NGETINT 1497
DPLN(ICHLNO) = DHYD*CINCH(IUNITS)	NGETINT 1498
ROUT(ICHLNO) = RO*CINCH(IUNITS)	NGETINT 1499
RIN(ICHLNO) = RI*CINCH(IUNITS)	NGETINT 1500

C		NGETINT 1501
C	NOW, /GEO/ IS READ, NGEOTIMES, AND THE DATA STORED IN CHANL ARRAY.	NGETINT 1502
C		NGETINT 1503
C		NGETINT 1504
C	ISBLOK = THE STARTING POINT IN CHANL ARRAY FOR CHANNEL ICHLNO DATA	NGETINT 1505
C		NGETINT 1506
C	FIRST, STORE THE SINGLE VALUED DATA	NGETINT 1507
C		NGETINT 1508
	CHANL (ISBLOK) = CD	NGETINT 1509
	CHANL (ISBLOK+1) = ALPHA	NGETINT 1510
	CHANL (ISBLOK+2) = BETA	NGETINT 1511
	CHANL (ISBLOK+3) = DELTA	NGETINT 1512
	CHANL (ISBLOK+4) = EPS	NGETINT 1513
	CHANL (ISBLOK+6) = ADUMP*CINCH(IUNITS)**2	NGETINT 1514
	CHANL (ISBLOK+7) = SPAN*CINCH(IUNITS)	NGETINT 1515
	CHANL (ISBLOK+8) = BTA	NGETINT 1516
	CHANL (ISBLOK+9) = DLTME	NGETINT 1517
	CHANL (ISBLOK+10) = TEPS	NGETINT 1518
C		NGETINT 1519
C	THEN THE ARRAYS ARE STORED:	NGETINT 1520
C		NGETINT 1521
C	THE FOLLOWING ARE STORED BY NODE NUMBER:	NGETINT 1522
C	THK (TAU), TDLX (DLX)	NGETINT 1523
C	THE REST ARE STORED BY STATION NUMBER:	NGETINT 1524
C	TDHJ (DHJ), TDHF (DHF), TXN (XN), TRR (RR),	NGETINT 1525
C	TDP (DP), TSP (SP),	NGETINT 1526
C	(AKC), (AKW), IHCT (IHC).	NGETINT 1527
C		NGETINT 1528
	ITHKZ = ISBLOK + 14	NGETINT 1529
	ITDLXZ = ISBLOK + 14 + NODST	NGETINT 1530
	ITDHJZ = ISBLOK + 14 + 2*NODST	NGETINT 1531
	ITDHFZ = ISBLOK + 14 + 2*NODST + NSTA	NGETINT 1532
	ITXNZ = ISBLOK + 14 + 2*NODST + 2*NSTA	NGETINT 1533
	ITERZ = ISBLOK + 14 + 2*NODST + 3*NSTA	NGETINT 1534
	ITDPZ = ISBLOK + 14 + 2*NODST + 4*NSTA	NGETINT 1535
	ITSPZ = ISBLOK + 14 + 2*NODST + 5*NSTA	NGETINT 1536
	IEND = ISBLOK + 14 + 2*NODST + 8*NSTA	NGETINT 1537
	THK (1) = 0.0	NGETINT 1538
	DO 170 I = 1, NGEOTIMES	NGETINT 1539
	ISTB = 0	NGETINT 1540
	READ (5, GEO)	NGETINT 1541
	IF (THK (1).LE.0.0) THK (1) = .0001*THK (2)	NGETINT 1542
	IF (TDLX (1).GT.2.0*TDLX (4).OR.TDLX (4).GT.1.2*TDLX (1))	NGETINT 1543
	Z WRITE (8, 136) ICHLNO, ISTA	NGETINT 1544
136	FORMAT (/ ' CHANNEL ', I2, ' , STATION ', I3,	NGETINT 1545
	Z ' , ---TDLX VALUES DO NOT LOOK RIGHT')	NGETINT 1546
	IF (TDHJ.GT.0..AND.TXN.LT.1.1*TDHJ) WRITE (8, 137) ICHLNO, ISTA	NGETINT 1547
137	FORMAT (/ ' CHANNEL ', I2, ' , STATION ', I3,	NGETINT 1548
	Z ' , ---HOLE SPACING AND DIAMETER DO NOT LOOK RIGHT')	NGETINT 1549
	IF (ISTB.EQ.0) ISTB = ISTA	NGETINT 1550
	DO 165 J = ISTA, ISTB, 2	NGETINT 1551
C		NGETINT 1552
C	J REPRESENTS THE STATION NUMBER IN THIS CASE	NGETINT 1553
C		NGETINT 1554
	JSENS = J - 2*(J/2)	NGETINT 1555
C		NGETINT 1556
C	JSENS = 0 INDICATES THAT STATION NO. IS EVEN AND STATION IS ON	NGETINT 1557
C	SUCTION SIDE	NGETINT 1558
C	JSENS = 1 INDICATES THAT STATION NO. IS ODD AND STATION IS ON	NGETINT 1559
C	PRESSURE SIDE	NGETINT 1560

C	IARG = ITDHJZ + J	NGETINT 1561
	CHANL(IARG) = TDHJ*CINCH(IUNITS)	NGETINT 1562
	IARG = ITDHPZ + J	NGETINT 1563
	CHANL(IARG) = TDHP*CINCH(IUNITS)	NGETINT 1564
	IARG = ITXNZ + J	NGETINT 1565
	CHANL(IARG) = TXN*CINCH(IUNITS)	NGETINT 1566
	IARG = ITRRZ + J	NGETINT 1567
	CHANL(IARG) = TRR*CINCH(IUNITS)	NGETINT 1568
	IARG = ITDPZ + J	NGETINT 1569
	CHANL(IARG) = TDP*CINCH(IUNITS)	NGETINT 1570
	IARG = ITSPZ + J	NGETINT 1571
	CHANL(IARG) = TSP*CINCH(IUNITS)	NGETINT 1572
	IARG = IIHCTZ + J	NGETINT 1573
	INDCHN(IARG) = IHCT	NGETINT 1574
	NODOUT = 5*J - 4	NGETINT 1575
C		NGETINT 1576
C	NODOUT IS THE NODE NO. ON THE OUTSIDE SURFACE AT STATION J	NGETINT 1577
C	5 IS THE NUMBER OF NODES AT STATION J	NGETINT 1578
C		NGETINT 1579
145	CONTINUE	NGETINT 1580
	LOCA = ITDLXZ + NODOUT	NGETINT 1581
	IF (TDLX(3).LE.0.) GO TO 155	NGETINT 1582
	DO 150 L = 1,5	NGETINT 1583
	LOCAL = LOCA + L - 1	NGETINT 1584
150	CHANL(LOCAL) = TDLX(L)*CINCH(IUNITS)	NGETINT 1585
	GO TO 160	NGETINT 1586
155	CHANL(LOCA) = TDLX(1)*CINCH(IUNITS)	NGETINT 1587
	CHANL(LOCA+3) = TDLX(4)*CINCH(IUNITS)	NGETINT 1588
	AA = TDLX(1)	NGETINT 1589
	B = (TDLX(4)-TDLX(1))/(THK(1)+THK(2))	NGETINT 1590
	CHANL(LOCA+1) = (AA + B*THK(1))*CINCH(IUNITS)	NGETINT 1591
	CHANL(LOCA+2) = (AA + B*(THK(1)+THK(2)/2.))*CINCH(IUNITS)	NGETINT 1592
	CHANL(LOCA+4) = (AA + B*(THK(1)+THK(2)+THK(3)/2.))*CINCH(IUNITS)	NGETINT 1593
160	CONTINUE	NGETINT 1594
	LOCA = ITHKZ + NODOUT	NGETINT 1595
	CHANL(LOCA) = THK(1)*CINCH(IUNITS)	NGETINT 1596
	CHANL(LOCA+2) = THK(2)*CINCH(IUNITS)	NGETINT 1597
	CHANL(LOCA+4) = THK(3)*CINCH(IUNITS)	NGETINT 1598
165	CONTINUE	NGETINT 1599
170	CONTINUE	NGETINT 1600
175	CONTINUE	NGETINT 1601
C		NGETINT 1602
C---	CONVERT UNITS ON BC DATA	NGETINT 1603
C		NGETINT 1604
	IF (IUNITS.EQ.2) GO TO 300	NGETINT 1605
	NTBC = 1	NGETINT 1606
	DO 205 I = 2,50	NGETINT 1607
	IF (BCTIME(I).LE.0.0) GO TO 210	NGETINT 1608
205	NTBC = NTBC + 1	NGETINT 1609
210	NSETS = NBCS*NSLICE*NTBC	NGETINT 1610
	NSETP = NBCE*NSLICE*NTBC	NGETINT 1611
C		NGETINT 1612
	DO 215 I = 1,NSETS	NGETINT 1613
	PCXS(I) = BCXS(I)*CINCH(1)	NGETINT 1614
	BCHGS(I) = BCHGS(I)*CHTC(1)	NGETINT 1615
	BCTGS(I) = BCTGS(I)*CTMPF(1) - 460.	NGETINT 1616
	BCQGS(I) = BCQGS(I)*CHFLX(1)	NGETINT 1617
215	BCPGS(I) = BCPGS(I)*CPRSR(1)	NGETINT 1618
C		NGETINT 1619
		NGETINT 1620

DO 220 I = 1,NSETP	NGETINT 1621
BCXP(I) = BCXP(I)*CINCH(1)	NGETINT 1622
BCHGP(I) = BCHGP(I)*CHTC(1)	NGETINT 1623
BCTGP(I) = BCTGP(I)*CTMPF(1) - 460.	NGETINT 1624
BCQGP(I) = BCQGP(I)*CHFLX(1)	NGETINT 1625
220 BCPGP(I) = BCPGP(I)*CPRSR(1)	NGETINT 1626
C	NGETINT 1627
NSET = NSTA*NTBC	NGETINT 1628
DO 225 I = 1,NSET	NGETINT 1629
RHOVG(I) = RHOVG(I)*CRHOVG(1)	NGETINT 1630
THUBIN(I) = THUBIN(I)*CTMPF(1) - 460.	NGETINT 1631
QHUBIN(I) = QHUBIN(I)*CHFLX(1)	NGETINT 1632
TTIPIN(I) = TTIPIN(I)*CTMPF(1) - 460.	NGETINT 1633
225 QTIPIN(I) = QTIPIN(I)*CHFLX(1)	NGETINT 1634
C	NGETINT 1635
NSET = NSLICE*NTBC	NGETINT 1636
DO 230 I = 1,NSET	NGETINT 1637
230 PEX(I) = PEX(I)*CPRSR(IUNITS)	NGETINT 1638
C	NGETINT 1639
DO 235 I = 1,49,2	NGETINT 1640
TTIO(I) = TTIO(I)*CTMPF(IUNITS) - 460.	NGETINT 1641
235 PTIO(I) = PTIO(I)*CPRSR(IUNITS)	NGETINT 1642
WPLEN = WPLEN*CMSFL(IUNITS)	NGETINT 1643
RHOC = RHOC*CDEN(IUNITS)	NGETINT 1644
RHOM = RHOM*CDEN(IUNITS)	NGETINT 1645
SPHTC = SPHTC*CSPHT(IUNITS)	NGETINT 1646
SPHTM = SPHTM*CSPHT(IUNITS)	NGETINT 1647
C	NGETINT 1648
DO 280 I = 1,19,2	NGETINT 1649
AKCTBL(I) = AKCTBL(I)*CTMPF(IUNITS) - 460.	NGETINT 1650
AKCTBL(I+1) = AKCTBL(I+1)*CTCON(IUNITS)	NGETINT 1651
AKWTBL(I) = AKWTBL(I)*CTMPF(IUNITS) - 460.	NGETINT 1652
280 AKWTBL(I+1) = AKWTBL(I+1)*CTCON(IUNITS)	NGETINT 1653
C	NGETINT 1654
300 CONTINUE	NGETINT 1655
C	NGETINT 1656
C	NGETINT 1657
IF (IFILM.NE.2) GO TO 320	NGETINT 1658
DO 310 I = 1,NSTA	NGETINT 1659
310 RHOVGA(I) = RHOVG(I)	NGETINT 1660
C	NGETINT 1661
320 CONTINUE	NGETINT 1662
C	NGETINT 1663
C IF INEDIT .GT. 0, PRINT AN INPUT EDIT	NGETINT 1664
C	NGETINT 1665
DO 180 I = 1,NSLICE	NGETINT 1666
180 CALL INPRT(I,INEDIT)	NGETINT 1667
185 CONTINUE	NGETINT 1668
RETURN	NGETINT 1669
END	NGETINT 1670
C----SOURCE.NHCFRCT	NHCFRCT 1671
FUNCTION HCFRCD(IS,LCOOL,LIN)	NHCFRCT 1672
C	NHCFRCT 1673
C- SOURCE.NHCFRCT----	NHCFRCT 1674
C	NHCFRCT 1675
COMMON /TCO/ ADUMP, BTA, CD, CP, TOG,	NHCFRCT 1676
Z GAM, PIM, R, SPAN,	NHCFRCT 1677
Z WDUMP, WIM, AKC(15,80), AKW(15,80),	NHCFRCT 1678
Z A(400), AJET(80), AM2(80), CNUM(80),	NHCFRCT 1679
Z DH(80), DHF(80), DHJ(80),	NHCFRCT 1680

```

Z      DLX(400),  PF(80),    HC(80),    HG(80),    NHCPRCT 1681
Z      P(2,15,80),PEXIT(15), PUMP(80),   OG(80),    NHCPRCT 1682
Z      QSNK(80),  RR(80),    S(15),     T(2,15,400), NHCPRCT 1683
Z      TG(80),   TAU(400),   WFC(80),    NHCPRCT 1684
Z      WJ(15,80), WCROS(2,15,80), XN(80),   NHCPRCT 1685
Z      ICOR,     IFILM,     IHUB,       ITIP,     NHCPRCT 1686
Z      ISBLOK,   ISLICE,    NBLKSZ,    NSLICE,    NHCPRCT 1687
Z      NFWD,     NSTA,      IHC(80)     NHCPRCT 1688
C      NHCPRCT 1689
C      COMPUTE TURBULENT HEAT TRANSFER COEFFICIENT IN CHANNEL FLOW: NHCPRCT 1690
C      NU = .023*( RE**.8 )*( PD**.333 ) NHCPRCT 1691
C      NHCPRCT 1692
100    CONTINUE NHCPRCT 1693
      TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2. NHCPRCT 1694
      CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU) NHCPRCT 1695
      FE = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(A(LCOOL)*XMU) NHCPRCT 1696
      HCFRCD = .023*12.*(C/DH(IS))*(RE**.8)*(PD**.333) NHCPRCT 1697
200    CONTINUE NHCPRCT 1698
      RETURN NHCPRCT 1699
      END NHCPRCT 1700

C-----SOURCE.NHCOOLT NHCoolT 1701
      SUBROUTINE HCOOL(JS) NHCoolT 1702
C      NHCoolT 1703
C- SOURCE.NHCOOLT---- NHCoolT 1704
C      NHCoolT 1705
C      NHCoolT 1706
      COMMON /IMPCOR/ CIMP1, CIMP2, CIMP3, CIMP4, CIMP5, CIMP6, CIMP7, NHCoolT 1707
Z      DIMP1, DIMP2, DIMP3, DIMP4, DIMP5, DIMP6 NHCoolT 1708
C      NHCoolT 1709
      COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80), NHCoolT 1710
Z      CPC(80), GAMC(80), DUMR1(80), DUMR2(80) NHCoolT 1711
C      NHCoolT 1712
      COMMON /TCO/ ADUMP, BTA, CD, CP, NHCoolT 1713
Z      GAM, FIM, R, SPAN, TOG, NHCoolT 1714
Z      WDUMP, WIM, AKC(15,80), AKW(15,80), NHCoolT 1715
Z      A(400), AJET(80), AM2(80), CNUM(80), NHCoolT 1716
Z      DH(80), DHP(80), DHJ(80), NHCoolT 1717
Z      DLX(400), PF(80), HC(80), HG(80), NHCoolT 1718
Z      P(2,15,80),PEXIT(15), PUMP(80), OG(80), NHCoolT 1719
Z      QSNK(80), RR(80), S(15), T(2,15,400), NHCoolT 1720
Z      TG(80), TAU(400), WFC(80), NHCoolT 1721
Z      WJ(15,80), WCROS(2,15,80), XN(80), NHCoolT 1722
Z      ICOR, IFILM, IHUB, ITIP, NHCoolT 1723
Z      ISBLOK, ISLICE, NBLKSZ, NSLICE, NHCoolT 1724
Z      NFWD, NSTA, IHC(80) NHCoolT 1725
C      NHCoolT 1726
      DIMENSION IGG(80), IRE(80), REJ(80), REJOVR(80) NHCoolT 1727
1      CONTINUE NHCoolT 1728
      TMP=TOG NHCoolT 1729
      CALL GASTBL (TMP,C,CP,GAM,PD,R,XMU) NHCoolT 1730
      CONDUCT = C NHCoolT 1731
      XMUTOG = XMU NHCoolT 1732
      PDTOG = PD NHCoolT 1733
      PI=3.14159 NHCoolT 1734
      IF (JS.GT.1) GO TO 101 NHCoolT 1735
      IF (ICOR.EQ.1) GO TO 101 NHCoolT 1736
      IF (WJ(ISLICE,JS).LE.0.0) GO TO 101 NHCoolT 1737
C      NHCoolT 1738
C--- LEADING EDGE HEAT TRANSFER CORRELATION FOR STATIONS FORWARD OF ICOR NHCoolT 1739
C--- CORRELATION OF METZGER ET AL, J. ENG. POWER, JULY 1969,PP 149-158 NHCoolT 1740

```

C		NHCOOLT 1741
5	NEND=ICOR+1	NHCOOLT 1742
	XS = 0.0	NHCOOLT 1743
	XP = 0.0	NHCOOLT 1744
	DO 50 J = 3,NEND,2	NHCOOLT 1745
	LINS = 5*(J-1) - 1	NHCOOLT 1746
	LINP = 5*J - 1	NHCOOLT 1747
	XS = XS + DLX(LINS)	NHCOOLT 1748
	XP = XP + DLX(LINP)	NHCOOLT 1749
50	CONTINUE	NHCOOLT 1750
55	XL = (XS + XP)/2.	NHCOOLT 1751
	IF (AJET(JS).GT.0.) GMASS = WJ(ISLICE,JS)/AJET(JS)	NHCOOLT 1752
60	BES = PI*DHJ(JS)**2/(4.*XN(JS))	NHCOOLT 1753
	DEH = 2.*BES	NHCOOLT 1754
65	REJ(JS) = 12.*3600.*GMASS*DEH/XMU	NHCOOLT 1755
	PROD = REJ(JS)**.27*(XL/BES)**.52	NHCOOLT 1756
70	STANMX = .355/PROD	NHCOOLT 1757
	HC(JS) = STANMX*CP*GMASS*144.*3600.	NHCOOLT 1758
C		NHCOOLT 1759
	IF (REJ(JS).LT.1150..OR.REJ(JS).GT.6300.) WRITE(6,75) REJ(JS)	NHCOOLT 1760
75	FORMAT(1H /'***WARNING*** LEADING EDGE IMPINGEMENT JET REYNOLDS ',	NHCOOLT 1761
Z	'NUMBER IS ',F8.1/' RANGE OF THE CORRELATION IS 1150',-	NHCOOLT 1762
Z	' < REJ < 6300')	NHCOOLT 1763
	ILEAD = ICOR - 1	NHCOOLT 1764
	IF (ILEAD.LT.2) GO TO 85	NHCOOLT 1765
	DO 80 I = 2,ILEAD	NHCOOLT 1766
	IF(WJ(ISLICE,I).GT.0.0) GO TO 90	NHCOOLT 1767
80	HC(I) = HC(JS)	NHCOOLT 1768
85	CONTINUE	NHCOOLT 1769
	GO TO 101	NHCOOLT 1770
90	WRITE(8,95) ICOR	NHCOOLT 1771
95	FORMAT(/' SOLUTION TERMINATED***TOO MANY ROWS OF IMPINGEMENT',	NHCOOLT 1772
Z	' HOLES FORWARD OF STATION',I3,'. HOLES ARE ',	NHCOOLT 1773
Z	' ALLOWED ONLY AT STATION 1.')	NHCOOLT 1774
	STOP	NHCOOLT 1775
C		NHCOOLT 1776
	C--KIRCHER-TABAKOFF CORRELATION, IMPINGEMENT WITH CROSS FLOW	NHCOOLT 1777
	C--ICOR = STATION NUMBER APPLICATION OF THIS CORRELATION BEGINS	NHCOOLT 1778
C		NHCOOLT 1779
101	IGGC = 0	NHCOOLT 1780
	IREC = 0	NHCOOLT 1781
	ISTRT=ICOR	NHCOOLT 1782
	IF (JS.GT.1) ISTRT= 1	NHCOOLT 1783
C		NHCOOLT 1784
	IF (CIMP1.NE.0.0) GO TO 400	NHCOOLT 1785
	DO 130 I = ISTRT,NPND	NHCOOLT 1786
	WC = ABS(WCROS(2,ISLICE,I))	NHCOOLT 1787
	II = 5*I	NHCOOLT 1788
	REJ(I) = 0.0	NHCOOLT 1789
	IF (IHC(I).EQ.1) GO TO 103	NHCOOLT 1790
	LCOOL = 5*I	NHCOOLT 1791
	LIN = LCOOL - 1	NHCOOLT 1792
	HC(I) = HCPRCD(I,LCOOL,LIN)	NHCOOLT 1793
	GO TO 130	NHCOOLT 1794
103	CONTINUE	NHCOOLT 1795
	IF (AJET(I).EQ.0.0) GO TO 128	NHCOOLT 1796
	IF (WJ(ISLICE,I).LE.0.) GO TO 128	NHCOOLT 1797
	TMP=(T(2,ISLICE,LIN)+T0G)/2.	NHCOOLT 1798
	CALL GASTBL (TMP,C,CP,GAM,PD,R,XMU)	NHCOOLT 1799
	CONDCT = C	NHCOOLT 1800

	XHUTOG = XHU	NHCOOLT 1801
	PDTOG = PD	NHCOOLT 1802
105	REJ(I) = WJ(ISLICE,I)/AJET(I)*DHJ(I)/(XHUTOG/3600.)*12.0	NHCOOLT 1803
	GG = (WC/A(II))/(WJ(ISLICE,I)/AJET(I))	NHCOOLT 1804
	IF (GG.LE.2.0) GO TO 110	NHCOOLT 1805
	IGGC = IGGC + 1	NHCOOLT 1806
	IGG(IGGC) = I	NHCOOLT 1807
110	CONTINUE	NHCOOLT 1808
	IF (REJ(I).GE.300.0.AND.REJ(I).LE.3.E4) GO TO 115	NHCOOLT 1809
	IREC = IREC + 1	NHCOOLT 1810
	IRE(IREC) = I	NHCOOLT 1811
115	CONTINUE	NHCOOLT 1812
	IF (REJ(I).LT.3000.) GO TO 120	NHCOOLT 1813
	AM = -.002517*(XN(I)/DHJ(I))**2+.068485*XN(I)/DHJ(I)+.506994	NHCOOLT 1814
	HC(I) = REJ(I)**AM	NHCOOLT 1815
	HC(I) = HC(I)*EXP(.02596*(XN(I)/DHJ(I))**2-.8259*XN(I)/DHJ(I)+.3985)	NHCOOLT 1816
	HC(I) = HC(I)/(1+.4696*	NHCOOLT 1817
	Z ((WC/A(II))/(WJ(ISLICE,I)/AJET(I))*TAU(II)/DHJ(I))**.965)	NHCOOLT 1818
	GO TO 125	NHCOOLT 1819
120	AM = -.001452*(XN(I)/DHJ(I))**2+.042838*(XN(I)/DHJ(I))+.516548	NHCOOLT 1820
	HC(I) = REJ(I)**AM	NHCOOLT 1821
	HC(I) = HC(I)*EXP(.0126*(XN(I)/DHJ(I))**2-.5106*XN(I)/DHJ(I)-.2057)	NHCOOLT 1822
	HC(I) = HC(I)/(1+.4215*	NHCOOLT 1823
	Z ((WC/A(II))/(WJ(ISLICE,I)/AJET(I))*TAU(II)/DHJ(I))**.58)	NHCOOLT 1824
125	CONTINUE	NHCOOLT 1825
	HC(I) = HC(I)*CONDCT/DHJ(I)*12.0*PDTOG**.33*(TAU(II)/DHJ(I))**.091	NHCOOLT 1826
	GO TO 130	NHCOOLT 1827
C		NHCOOLT 1828
128	CONTINUE	NHCOOLT 1829
	IF (I.GT.2) HC(I) = HC(I-2)	NHCOOLT 1830
	IF (I.EQ.2) HC(I) = HC(1)	NHCOOLT 1831
	IF (I.EQ.1) HC(I) = HC(3)	NHCOOLT 1832
130	CONTINUE	NHCOOLT 1833
	IST = NFWD + 1	NHCOOLT 1834
	DO 150 I = IST,NSTA,2	NHCOOLT 1835
	IF (IHC(I).NE.1) GO TO 155	NHCOOLT 1836
150	HC(I) = HC(I-2)	NHCOOLT 1837
155	IST = NFWD+2	NHCOOLT 1838
	DO 160 I = IST,NSTA,2	NHCOOLT 1839
	IF (IHC(I).NE.1) GO TO 165	NHCOOLT 1840
160	HC(I) = HC(I-2)	NHCOOLT 1841
165	CONTINUE	NHCOOLT 1842
	IF (IGGC.GT.0) WRITE(6,140) (IGG(I),I=1,IGGC)	NHCOOLT 1843
	DO 132 I = 1,IREC	NHCOOLT 1844
	ISTATN = IRE(I)	NHCOOLT 1845
	REJOVR(I) = REJ(ISTATN)	NHCOOLT 1846
132	CONTINUE	NHCOOLT 1847
	IF (IREC.GT.0) WRITE(6,145) (IRE(I),REJOVR(I),I=1,IREC)	NHCOOLT 1848
135	CONTINUE	NHCOOLT 1849
140	FORMAT(1H /' ***** WARNING ***** RATIO OF CROSSFLOW TO ',	NHCOOLT 1850
	Z 'JET-FLOW IS OUT OF THE RANGE OF ',	NHCOOLT 1851
	Z 'THE CORRELATION AT THE FOLLOWING STATIONS: '/23X,20(I4,','))	NHCOOLT 1852
145	FORMAT(1H /' ***** WARNING ***** JET REYNOLD'S NUMBER IS ',	NHCOOLT 1853
	Z 'OUT OF THE RANGE OF THE CORRELATION ',	NHCOOLT 1854
	Z 'AT THE FOLLOWING STATIONS: '/1X,8('**',I2,'--',F8.1,'**'))	NHCOOLT 1855
	DO 301 I = 1,NFWD	NHCOOLT 1856
	DUMR2(I) = REJ(I)	NHCOOLT 1857
301	CONTINUE	NHCOOLT 1858
	RETURN	NHCOOLT 1859
C		NHCOOLT 1860

C--- GENERAL CORRELATION FOR IMPINGEMENT WITH CROSSFLOW IS EVALUATED HERE	NHCOOLT 1861
C	NHCOOLT 1862
C--- FORM OF CORRELATION IS:	NHCOOLT 1863
C--- ST = CIMP1*(GG**CIMP2)*(GI**CIMP3)*((Z/D)**CIMP4)	NHCOOLT 1864
C	NHCOOLT 1865
C--- WHERE GG IS THE MASS FLUX RATIO, FREE STREAM TO JET, AND	NHCOOLT 1866
C--- GI IS THE MOMENTUM FLUX RATIO.	NHCOOLT 1867
C	NHCOOLT 1868
400 CONTINUE	NHCOOLT 1869
DO 450 I = ISTRT,NFWD	NHCOOLT 1870
WC = ABS(WCROS(2,ISLICE,I))	NHCOOLT 1871
II = 5*I	NHCOOLT 1872
ROINVJ = R*TOG/(144.*P(2,ISLICE,I))	NHCOOLT 1873
ROINVC = R*T(2,ISLICE,II)/(144.*P(2,ISLICE,I))	NHCOOLT 1874
REJ(I) = 0.0	NHCOOLT 1875
IF (IHC(I).EQ.1) GO TO 403	NHCOOLT 1876
LCOOL = 5*I	NHCOOLT 1877
LIN = LCOOL - 1	NHCOOLT 1878
HC(I) = HCFRCD(I,LCOOL,LIN)	NHCOOLT 1879
GO TO 450	NHCOOLT 1880
403 CONTINUE	NHCOOLT 1881
IF (AJET(I).EQ.0.0) GO TO 445	NHCOOLT 1882
IF (WJ(ISLICE,I).LE.0.) GO TO 445	NHCOOLT 1883
TMP=(T(2,ISLICE,LIN)+TOG)/2.	NHCOOLT 1884
CALL GASTBL (TMP,C,CP,GAM,PD,R,XMU)	NHCOOLT 1885
CONDC = C	NHCOOLT 1886
XMUTOG = XMU	NHCOOLT 1887
PDTOG = PD	NHCOOLT 1888
405 REJ(I)=WJ(ISLICE,I)/AJET(I)*DHJ(I)/(XMUTOG/3600.)*12.0	NHCOOLT 1889
GG=(WC/A(II))/(WJ(ISLICE,I)/AJET(I))	NHCOOLT 1890
GI = ((WC/A(II))**2*ROINVC)/((WJ(ISLICE,I)/AJET(I))**2*ROINVJ)	NHCOOLT 1891
ZOVERD = TAU(II)/DHJ(I)	NHCOOLT 1892
XOVERD = XN(I)/DHJ(I)	NHCOOLT 1893
ST = CIMP1*(GG**CIMP2)*(GI**CIMP3)*(ZOVERD**CIMP4)	NHCOOLT 1894
Z	NHCOOLT 1895
* (XOVERD**CIMP5)*(REJ(I)**CIMP6)*(PDTOG**CIMP7)	NHCOOLT 1896
HC(I) = 144.*3600.*ST*CP*WJ(ISLICE,I)/AJET(I)	NHCOOLT 1897
GO TO 450	NHCOOLT 1898
C	NHCOOLT 1899
445 CONTINUE	NHCOOLT 1900
IF (I.GT.2) HC(I) = HC(I-2)	NHCOOLT 1901
IF (I.EQ.2) HC(I) = HC(1)	NHCOOLT 1902
IF (I.EQ.1) HC(I) = HC(3)	NHCOOLT 1903
450 CONTINUE	NHCOOLT 1904
IST = NFWD+1	NHCOOLT 1905
DO 460 I = IST,NSTA,2	NHCOOLT 1906
IF (IHC(I).NE.1) GO TO 465	NHCOOLT 1907
460 HC(I) = HC(I-2)	NHCOOLT 1908
465 IST = NFWD+2	NHCOOLT 1909
DO 470 I = IST,NSTA,2	NHCOOLT 1910
IF (IHC(I).NE.1) GO TO 475	NHCOOLT 1911
470 HC(I) = HC(I-2)	NHCOOLT 1912
475 CONTINUE	NHCOOLT 1913
C	NHCOOLT 1914
DO 485 I = 1,NFWD	NHCOOLT 1915
DUMR2(I) = REJ(I)	NHCOOLT 1916
485 CONTINUE	NHCOOLT 1917
RETURN	NHCOOLT 1918
END	
C----SOURCE.NHCPINT	NHCPINT 1919
SUBROUTINE HCPINS (IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EPAREA)	NHCPINT 1920

C		NHCPINT 1921
C-	SOURCE.NHCPINT----	NHCPINT 1922
C		NHCPINT 1923
	COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),	NHCPINT 1924
Z	CPC(80), GAMC(80), DUMR1(80), DUMR2(80)	NHCPINT 1925
C		NHCPINT 1926
	COMMON /TCO/ ADUMP, BTA, CD, CP, TOG,	NHCPINT 1927
Z	GAM, PIN, R, SPAN,	NHCPINT 1928
Z	WDUMP, WIM, AKC(15,80), AKW(15,80),	NHCPINT 1929
Z	A(400), AJET(80), AH2(80), CNUM(80),	NHCPINT 1930
Z	DH(80), DHF(80), DHJ(80),	NHCPINT 1931
Z	DLX(400), FF(80), HC(80), HG(80),	NHCPINT 1932
Z	P(2,15,80), PEXIT(15), PUMP(80), QG(80),	NHCPINT 1933
Z	QSNK(80), RR(80), S(15), T(2,15,400),	NHCPINT 1934
Z	TG(80), TAU(400), WFC(80),	NHCPINT 1935
Z	WJ(15,80), WCROS(2,15,80), XN(80),	NHCPINT 1936
Z	ICOR, IPILM, IHUB, ITIP,	NHCPINT 1937
Z	ISBLOK, ISLICE, NBLKSZ, NSLICE,	NHCPINT 1938
Z	NFWD, NSTA, IHC(80)	NHCPINT 1939
C		NHCPINT 1940
	DIMENSION EPAREA(80), DELTAN(15)	NHCPINT 1941
C		NHCPINT 1942
C	COMPUTE THE HEAT TRANSFER COEFFICIENT AND EFFECTIVENESS FOR A	NHCPINT 1943
C	TRIANGULAR ARRAY OF PIN FINS	NHCPINT 1944
C		NHCPINT 1945
C	WHERE DP IS PIN DIAMETER IN INCHES AND SP IS PIN SPACING IN INCHES	NHCPINT 1946
C		NHCPINT 1947
	VDP = DP(IS)	NHCPINT 1948
	VSP = SP(IS)	NHCPINT 1949
100	CONTINUE	NHCPINT 1950
	TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.	NHCPINT 1951
	CALL GASTBL(TMP,C,CP,GAM,PD,R,XHU)	NHCPINT 1952
C--	NO. OF PINS AT THIS STATION IS:	NHCPINT 1953
	PINS = SPAN*DLX(LCOOL)/(.86603*VSP**2)	NHCPINT 1954
C--	AVERAGE LENGTH OF PINS:	NHCPINT 1955
	SLP = (TAU(LCOOL) + TAU(LCUP))/2.	NHCPINT 1956
	IF (IS.GT.NFWD.AND.IS.LE.NFWD+2) SLP = TAU(LCOOL)	NHCPINT 1957
C--	MINIMUM FLOW AREA:	NHCPINT 1958
	AMIN = SLP*SPAN*(VSP-VDP)/VSP	NHCPINT 1959
C--	TOTAL SURFACE AREA:	NHCPINT 1960
	AHTTR = 2.*DLX(LCOOL)*SPAN + 3.14159*PINS*(VDP*SLP-VDP**2/4.)	NHCPINT 1961
C--	CHANNEL HYDRAULIC DIAMETER:	NHCPINT 1962
	DH(IS) = 4.*AMIN*DLX(LCOOL)/AHTTR	NHCPINT 1963
	REDH = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(AMIN*XHU)	NHCPINT 1964
	TERM1 = -.89*(VSP/SLP)**.5075	NHCPINT 1965
	TERM2 = -3.094*VDP/VSP	NHCPINT 1966
	TERM3 = 4.143*EXP(TERM1 + TERM2)/(REDH**.2946)	NHCPINT 1967
105	CONTINUE	NHCPINT 1968
	HC(IS) = (12.*C/DH(IS))*(.023 + TERM3)*(REDH**.8)*(PD**.333)	NHCPINT 1969
	EML = SQRT(4.*HC(IS)*SLP**2/(AKW(ISLICE,IS)*VDP))	NHCPINT 1970
	EPTVNS = TANH(EML)/EML	NHCPINT 1971
C--	CHECK LOCATION OF HEAT FLOW SPLIT POINT IF THIS IS A TRAILING	NHCPINT 1972
C	EDGE REGION STATION	NHCPINT 1973
C		NHCPINT 1974
	IF (IS.LE.NFWD) GO TO 160	NHCPINT 1975
	TBAR = (T(2,ISLICE,LCOOLP)-T(2,ISLICE,LCOOL))/	NHCPINT 1976
Z	(T(2,ISLICE,LIN)-T(2,ISLICE,LCOOL))	NHCPINT 1977
	HYCOS = COSH(EML)	NHCPINT 1978
	HYSIN = SINH(EML)	NHCPINT 1979
	IF (HYCOS-TBAR.LT.HYSIN) GO TO 120	NHCPINT 1980

110	WRITE(6,110) LCOOLP,LIN,DELTAN(ISLICE)	NHCPINT 1981
	FORMAT(1H //' **** WARNING **** NODE',I3,	NHCPINT 1982
Z	' IS RECEIVING HEAT FROM NODE',I3,' THROUGH THE PINS.',	NHCPINT 1983
Z	' RESULTS ARE INVALID. DELTAN =',F7.4)	NHCPINT 1984
	GO TO 140	NHCPINT 1985
120	CONTINUE	NHCPINT 1986
	IF (HYCOS-TBAR.GT.0.) GO TO 130	NHCPINT 1987
	WRITE(6,110) LIN,LCOOLP,DELTAN(ISLICE)	NHCPINT 1988
	GO TO 140	NHCPINT 1989
130	CONTINUE	NHCPINT 1990
	XOVR = (HYCOS-TBAR)/HYSIN	NHCPINT 1991
	XOVR = ALOG((1.+XOVR)/(1.-XOVR))/(2.*EML)	NHCPINT 1992
140	CONTINUE	NHCPINT 1993
160	CONTINUE	NHCPINT 1994
	EPAREA(IS) = DLX(LIN)*SPAN	NHCPINT 1995
Z	- 3.14159*PINS*(VDP**2/4.- EPTVNS*VDP*SLP*XOVR)	NHCPINT 1996
	IF (IS.GT.NFWD) EPAREA(IS+1) = DLX(LCOOLP)*SPAN	NHCPINT 1997
Z	- 3.14159*PINS*(VDP**2/4.- EPTVNS*VDP*SLP*(1.-XOVR))	NHCPINT 1998
	IF (IS.LE.NFWD) EPAREA(IS) = DLX(LIN)*SPAN	NHCPINT 1999
Z	- 3.14159*PINS*(VDP**2/4.- EPTVNS*VDP*SLP)	NHCPINT 2000
170	CONTINUE	NHCPINT 2001
	RETURN	NHCPINT 2002
	END	NHCPINT 2003
C-----SOURCE.NINPRTT		NINPRTT 2004
	SUBROUTINE INPRT(ICHNL,INEDIT)	NINPRTT 2005
C		NINPRTT 2006
C-	SOURCE.NINPRTT----	NINPRTT 2007
C		NINPRTT 2009
	COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000),	NINPRTT 2009
Z	BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),	NINPRTT 2010
Z	BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),	NINPRTT 2011
Z	QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80),	NINPRTT 2012
Z	QTIPIN(400), QTIP(80), PHOVG(400), PEX(400),	NINPRTT 2013
Z	BCTIME(50), TTIO(50), PTIO(50), WPLEN,	NINPRTT 2014
Z	WSVST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCP	NINPRTT 2015
C		NINPRTT 2016
	COMMON /GAAS/ GS(200), NG	NINPRTT 2017
C		NINPRTT 2018
	COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),	NINPRTT 2019
Z	CPC(80), GAMC(80), DUMR1(80), DUMR2(80)	NINPRTT 2020
C		NINPRTT 2021
	COMMON /RADL/ APLN(15), DPLN(15), RIN(15), ROUT(15),	NINPRTT 2022
Z	PIN(15), TIN(15), W(15), WS	NINPRTT 2023
	COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000),	NINPRTT 2024
Z	IPLLOT, MD1, MD2, MD3, IADJIN, IWRITE	NINPRTT 2025
C		NINPRTT 2026
	COMMON /TCO/ ADUMP, BTA, CD, CP,	NINPRTT 2027
Z	GAM, PIN, R, SPAN, TOG,	NINPRTT 2028
Z	WDUMP, WIN, AKC(15,80), AKW(15,80),	NINPRTT 2029
Z	A(400), AJET(80), AM2(80), CNUM(80),	NINPRTT 2030
Z	DH(80), DHF(80), DHJ(80),	NINPRTT 2031
Z	DLX(400), PF(80), HC(80), HG(80),	NINPRTT 2032
Z	P(2,15,80), PEXIT(15), PUMP(80), QG(80),	NINPRTT 2033
Z	QSNK(80), RR(80), S(15), T(2,15,400),	NINPRTT 2034
Z	TG(80), TAU(400), WFC(80),	NINPRTT 2035
Z	WJ(15,80), WCROS(2,15,80), XN(80),	NINPRTT 2036
Z	ICOR, IPILM, IHUB, ITIP,	NINPRTT 2037
Z	ISBLOK, ISLICE, NBLKSZ, NSLICE,	NINPRTT 2038
Z	NFWD, NSTA, IHC(80)	NINPRTT 2039
C		NINPRTT 2040

	COMMON /TRANS/ RHOC,	RHOM,	SPHTC,	SPHTH,	NINPRTT 2041
Z	DLTYME,	TYME,	TEPS,	TYHMAI	NINPRTT 2042
C					NINPRTT 2043
	COMMON /UNITS/ CINCH(2), CHTC(2),	CHFLX(2),	CPRSR(2),	CHSFL(2),	NINPRTT 2044
Z	CTMPF(2), CTCON(2),	CDEN(2),	CSPHT(2),	CGASC(2),	NINPRTT 2045
Z	CVISC(2), CRHOVG(2), IUNITS				NINPRTT 2046
C					NINPRTT 2047
	DIMENSION DUM1(10), DUM2(10), DUM3(10), DUM4(10), DUM5(10), DUM6(10),				NINPRTT 2048
Z	DUM7(10), DUM8(10), DUM9(10), DUM53(10), DUM55(10)				NINPRTT 2049
	DIMENSION DUM10(10), DUM11(10), DUM12(10), DUM13(10), DUM14(10),				NINPRTT 2050
Z	DUM15(10), DUM16(10), DUM25(10), DUM52(10)				NINPRTT 2051
	DIMENSION DUM17(10), DUM18(10), DUM19(10), DUM20(10)				NINPRTT 2052
	DIMENSION NFLUID(200), HCAL(4), UL(2), UA(2)				NINPRTT 2053
	DATA HCAL/'INPG','CHAN','PINS','				NINPRTT 2054
	DATA UL/'CM','IN'				NINPRTT 2055
	DATA UA/'CM','IN*'				NINPRTT 2056
	CALL PREP(ICHNL,NZON,1)				NINPRTT 2057
C					NINPRTT 2058
C	INITIALIZE TEMPERATURE DISTRIBUTION (DEGREES R)				NINPRTT 2059
C					NINPRTT 2060
	I = ICHNL				NINPRTT 2061
	NODSF = 5*NPWD				NINPRTT 2062
	NODSTM = 5*NSTA - 4				NINPRTT 2063
	NODST = 5*NSTA				NINPRTT 2064
	DO 830 I1 = 5, NODSI, 5				NINPRTT 2065
	IS = I1/5				NINPRTT 2066
	LO = I1-4				NINPRTT 2067
	LJ = I1-3				NINPRTT 2068
	L = I1-2				NINPRTT 2069
	LI = I1-1				NINPRTT 2070
	T(2,I,LO) = .9*TG(IS)				NINPRTT 2071
	T(2,I,LI) = T(2,I,LO)/1.08				NINPRTT 2072
	T(2,I,LJ) = T(2,I,LO) - (T(2,I,LO) - T(2,I,LI))*TAU(LO)/(TAU(LO) +				NINPRTT 2073
Z	TAU(L))				NINPRTT 2074
Z	T(2,I,L) = T(2,I,LO) - (T(2,I,LO) - T(2,I,LI))*(TAU(LO) + TAU(L)/2.)/				NINPRTT 2075
	(TAU(LO) + TAU(L))				NINPRTT 2076
930	T(2,I,I1) = TTIO(1) + 460.				NINPRTT 2077
	ISTR = NODSF + 5				NINPRTT 2078
	DO 860 I1 = ISTR, NODSTM, 10				NINPRTT 2079
	T(2,I,I1) = T(2,I,NODSF)				NINPRTT 2080
	T(2,I,I1+5) = T(2,I,I1)				NINPRTT 2081
	DO 860 J = 1, 4				NINPRTT 2082
	IPJ = I1 + J				NINPRTT 2083
	IMJ = I1 + J - 5				NINPRTT 2084
	IUPP = NODSF + J - 5				NINPRTT 2085
	T(2,I,IPJ) = T(2,I,IUPP)				NINPRTT 2086
860	T(2,I,IMJ) = T(2,I,IUPP)				NINPRTT 2087
	DO 865 J = 1, NODST				NINPRTT 2088
865	T(1,I,J) = T(2,I,J)				NINPRTT 2089
C					NINPRTT 2090
	IF (ICHNL.GT.1) GO TO 94				NINPRTT 2091
	WRITE(6,408)				NINPRTT 2092
408	FORMAT(1H1,////,20X,'PROPERTY TABLES'///)				NINPRTT 2093
	WRITE(6,410)				NINPRTT 2094
410	FORMAT(1H , 'OUTER COATING EFFECTIVE THERMAL CONDUCTIVITY')				NINPRTT 2095
C					NINPRTT 2096
	IF (IUNITS.EQ.1) GO TO 420				NINPRTT 2097
C					NINPRTT 2098
	WRITE(6,412) (AKCTBL(I), I=1, 19, 2)				NINPRTT 2099
	WRITE(6,414) (AKCTBL(I), I=2, 20, 2)				NINPRTT 2100

412	FORMAT(/5X,'T, (F)',10X,10(F9.1))	NINPRTT 2101
414	FORMAT(5X,'K, (BTU/HR/FT/R)',10(F9.3))	NINPRTT 2102
	WRITE(6,416)	NINPRTT 2103
416	FORMAT(///// 'WALL METAL THERMAL CONDUCTIVITY')	NINPRTT 2104
	WRITE(6,412) (AKWTBL(I),I=1,19,2)	NINPRTT 2105
	WRITE(6,414) (AKWTBL(I),I=2,20,2)	NINPRTT 2106
C		NINPRTT 2107
	GO TO 445	NINPRTT 2108
C		NINPRTT 2109
420	CONTINUE	NINPRTT 2110
	DO 418 I = 1,19,2	NINPRTT 2111
	AKCTBL(I) = (AKCTBL(I)+460.)/1.8	NINPRTT 2112
	AKWTBL(I) = (AKWTBL(I)+460.)/1.8	NINPRTT 2113
	AKCTBL(I+1) = AKCTBL(I+1)/CTCON(1)	NINPRTT 2114
418	AKWTBL(I+1) = AKWTBL(I+1)/CTCON(1)	NINPRTT 2115
C		NINPRTT 2116
	WRITE(6,422) (AKCTBL(I),I=1,19,2)	NINPRTT 2117
	WRITE(6,424) (AKCTBL(I),I=2,20,2)	NINPRTT 2118
422	FORMAT(/5X,'T, (K)',4X,10(F9.1))	NINPRTT 2119
424	FORMAT(5X,'K, (W/H/K)',10(F9.3))	NINPRTT 2120
	WRITE(6,416)	NINPRTT 2121
	WRITE(6,422) (AKWTBL(I),I=1,19,2)	NINPRTT 2122
	WRITE(6,424) (AKWTBL(I),I=2,20,2)	NINPRTT 2123
C		NINPRTT 2124
C		NINPRTT 2125
	DO 448 I = 1,19,2	NINPRTT 2126
	AKCTBL(I) = 1.8*AKCTBL(I) - 460.	NINPRTT 2127
	AKWTBL(I) = 1.8*AKWTBL(I) - 460.	NINPRTT 2128
	AKCTBL(I+1) = AKCTBL(I+1)*CTCON(1)	NINPRTT 2129
448	AKWTBL(I+1) = AKWTBL(I+1)*CTCON(1)	NINPRTT 2130
C		NINPRTT 2131
445	CONTINUE	NINPRTT 2132
C		NINPRTT 2133
	WRITE(6,450)	NINPRTT 2134
450	FORMAT(1H ///// 'TABLE OF GAS PROPERTIES')	NINPRTT 2135
	NGS = NG	NINPRTT 2136
	IF (NG.GT.10) NGS = 10	NINPRTT 2137
C		NINPRTT 2138
	IF (IUNITS.EQ.1) GO TO 470	NINPRTT 2139
C		NINPRTT 2140
	WRITE(6,452) (GS(J),J=1,NGS)	NINPRTT 2141
452	FORMAT(/5X,'TEMPERATURE (F)',10(F9.1))	NINPRTT 2142
	L = NG + 1	NINPRTT 2143
	LE = NG + NGS	NINPRTT 2144
	WRITE(6,454) (GS(J),J=L,LE)	NINPRTT 2145
454	FORMAT(5X,'K, (BTU/HR/FT/R)',10(F9.5))	NINPRTT 2146
	L = 2*NG + 1	NINPRTT 2147
	LE = 2*NG + NGS	NINPRTT 2148
	WRITE(6,456) (GS(J),J=L,LE)	NINPRTT 2149
456	FORMAT(5X,'CP, (BTU/LBM/R)',10(F9.5))	NINPRTT 2150
	L = 3*NG + 1	NINPRTT 2151
	LE = 3*NG + NGS	NINPRTT 2152
	WRITE(6,458) (GS(J),J=L,LE)	NINPRTT 2153
458	FORMAT(5X,'PRANDTL NUMBER',10(F9.5))	NINPRTT 2154
	L = 4*NG + 1	NINPRTT 2155
	LE = 4*NG + NGS	NINPRTT 2156
	WRITE(6,460) (GS(J),J=L,LE)	NINPRTT 2157
460	FORMAT(5X,'VIS. (LBM/FT/HR)'10(F9.5))	NINPRTT 2158
C		NINPRTT 2159
	GO TO 90	NINPRTT 2160

C		NINPRTT 2161
C		NINPRTT 2162
470	CONTINUE	NINPRTT 2163
	DO 471 J = 1,NGS	NINPRTT 2164
471	DUM1(J) = (GS(J)+460.)/1.8	NINPRTT 2165
	WRITE(6,472) (DUM1(J),J=1,NGS)	NINPRTT 2166
472	FORMAT(/5X,'TEMPERATURE (K)',10(F9.1))	NINPRTT 2167
	L = NG + 1	NINPRTT 2168
	LE = NG + NGS	NINPRTT 2169
	JI = 0	NINPRTT 2170
	DO 473 J = L,LE	NINPRTT 2171
	JI = JI+1	NINPRTT 2172
473	DUM1(JI) = GS(J)/CTCON(1)	NINPRTT 2173
	WRITE(6,474) (DUM1(J),J=1,JI)	NINPRTT 2174
474	FORMAT(5X,'K, (W/M/K)',10(F9.5))	NINPRTT 2175
	L = 2*NG + 1	NINPRTT 2176
	LE = 2*NG + NGS	NINPRTT 2177
	JI = 0	NINPRTT 2178
	DO 475 J = L,LE	NINPRTT 2179
	JI = JI+1	NINPRTT 2180
475	DUM1(JI) = GS(J)/CSPHT(1)	NINPRTT 2181
	WRITE(6,476) (DUM1(J),J=1,JI)	NINPRTT 2182
476	FORMAT(5X,'CP, (J/KG/K)',10(F9.2))	NINPRTT 2183
	L = 3*NG + 1	NINPRTT 2184
	LE = 3*NG + NGS	NINPRTT 2185
	WRITE(6,478) (GS(J),J=L,LE)	NINPRTT 2186
478	FORMAT(5X,'PRANDTL NUMBER',10(F9.5))	NINPRTT 2187
	L = 4*NG + 1	NINPRTT 2188
	LE = 4*NG + NGS	NINPRTT 2189
	JI = 0	NINPRTT 2190
	DO 479 J = L,LE	NINPRTT 2191
	JI = JI+1	NINPRTT 2192
479	DUM1(JI) = GS(J)/CVISC(1)	NINPRTT 2193
	WRITE(6,480) (DUM1(J),J=1,JI)	NINPRTT 2194
480	FORMAT(5X,'VIS. (N S/M**2)'10(F9.5))	NINPRTT 2195
C		NINPRTT 2196
90	CONTINUE	NINPRTT 2197
	IF (INEDIT.EQ.0) GO TO 350	NINPRTT 2198
C		NINPRTT 2199
C--	LIST OUT THE INPUT HOT GAS BOUNDARY CONDITIONS--	NINPRTT 2200
C		NINPRTT 2201
C--	MNBC IS THE MAX OF (NBCS & NBCP)	NINPRTT 2202
C		NINPRTT 2203
	MNBC = NBCS	NINPRTT 2204
	IF (MNBC.LT.NBCP) MNBC=NBCP	NINPRTT 2205
	NTIMES = 1	NINPRTT 2206
C--	NTIMES IS THE NUMBER OF TIME STEPS IN BC TABLES	NINPRTT 2207
481	IF (BCTIME(NTIMES+1).LE.0.0) GO TO 482	NINPRTT 2208
	NTIMES = NTIMES + 1	NINPRTT 2209
	GO TO 481	NINPRTT 2210
C		NINPRTT 2211
482	CONTINUE	NINPRTT 2212
	WRITE(6,4820)	NINPRTT 2213
4820	FORMAT(1H1,40X,'HOT GAS BOUNDARY CONDITIONS'/)	NINPRTT 2214
	WRITE(6,483)	NINPRTT 2215
483	FORMAT(' *****SUCTION SIDE*****',22X,	NINPRTT 2216
	Z ' *****PRESSURE SIDE*****')	NINPRTT 2217
C		NINPRTT 2218
C--	SET THE NO. OF POINTS PER TIME STEP IN S&P BC ARRAYS	NINPRTT 2219
	NPRTS = NSLICE*NBCS	NINPRTT 2220

C	NPRTP = NSLICE*NBCP	NINPRTT 2221
	NL = 3	NINPRTT 2222
	DO 499 IT = 1,NTIMES	NINPRTT 2223
C		NINPRTT 2224
C--	SET THE NO. OF POINTS THAT PRECEDED TIME STEP 'IT'	NINPRTT 2225
	NPRCS = NPRTS*(IT-1)	NINPRTT 2226
	NPRCP = NPRTP*(IT-1)	NINPRTT 2227
C		NINPRTT 2228
C--	START THE LOOP THROUGH ALL SLICES	NINPRTT 2229
	DO 499 ISL = 1,NSLICE	NINPRTT 2230
C		NINPRTT 2231
C--	SET THE NO. OFF POINTS PRECEDING THIS SLICE	NINPRTT 2232
	NBFRS = NPRCS + NBCS*(ISL-1)	NINPRTT 2233
	NBFRP = NPRCP + NBCP*(ISL-1)	NINPRTT 2234
C		NINPRTT 2235
	NL = NL + 3 + MNBC	NINPRTT 2236
	IF (NL.LT.60) GO TO 4860	NINPRTT 2237
	NL = 3 + MNBC	NINPRTT 2238
	WRITE(6,4820)	NINPRTT 2239
	WRITE(6,483)	NINPRTT 2240
4860	CONTINUE	NINPRTT 2241
C		NINPRTT 2242
	IF (IT.EQ.1) WRITE(6,484) ISL	NINPRTT 2243
484	FORMAT(45X,'INITIAL STEADY STATE'/49X,'SLICE NO.',I2)	NINPRTT 2244
	IF (IT.GT.1) WRITE(6,485) BCTIME(IT), ISL	NINPRTT 2245
485	FORMAT(45X,'BCTIME =',F8.3,' SEC'/47X,'SLICE NO.',I2)	NINPRTT 2246
	WRITE(6,486)	NINPRTT 2247
486	FORMAT(4X,'X',7X,'HG',6X,'TG',10X,'QG',6X,'PG',	NINPRTT 2248
Z	26X,'X',7X,'HG',6X,'TG',10X,'QG',6X,'PG')	NINPRTT 2249
C		NINPRTT 2250
C--	HERE WE LOOP WITHIN A SLICE	NINPRTT 2251
C		NINPRTT 2252
	DO 499 IBC = 1,MNBC	NINPRTT 2253
	IF (IBC.GT.NBCS) GO TO 487	NINPRTT 2254
	J = NBFRS + IBC	NINPRTT 2255
	JXS = (ISL-1)*NBCS + IBC	NINPRTT 2256
	TBCXS = BCXS(JXS)/CINCH(IUNITS)	NINPRTT 2257
	TBCHGS = BCHGS(J)/CHTC(IUNITS)	NINPRTT 2258
	TBCTGS = BCTGS(J)	NINPRTT 2259
	IF (IUNITS.EQ.1) TBCTGS=(BCTGS(J)+460.)/1.8	NINPRTT 2260
	TBCQGS = BCQGS(J)/CHFLX(IUNITS)	NINPRTT 2261
	TBCPGS = BCPGS(J)/CPRSR(IUNITS)	NINPRTT 2262
	WRITE(6,489) TBCXS,TBCHGS,TBCTGS,TBCQGS,TBCPGS	NINPRTT 2263
489	FORMAT(2X,F6.2,2F8.1,F12.1,F8.1)	NINPRTT 2264
487	CONTINUE	NINPRTT 2265
	IF (IBC.GT.NBCP) GO TO 499	NINPRTT 2266
	J = NBFRP + IBC	NINPRTT 2267
	JXP = (ISL-1)*NBCP + IBC	NINPRTT 2268
	TBCXP = BCXP(JXP)/CINCH(IUNITS)	NINPRTT 2269
	TBCHGP = BCHGP(J)/CHTC(IUNITS)	NINPRTT 2270
	TBCTGP = BCTGP(J)	NINPRTT 2271
	IF (IUNITS.EQ.1) TBCTGP=(BCTGP(J)+460.)/1.8	NINPRTT 2272
	TBCQGP = BCQGP(J)/CHFLX(IUNITS)	NINPRTT 2273
	TBCPGP = BCPGP(J)/CPRSR(IUNITS)	NINPRTT 2274
	IF (IBC.LE.NBCS) WRITE(6,488) TBCXP,TBCHGP,TBCTGP,TBCQGP,TBCPGP	NINPRTT 2275
	IF (IBC.GT.NBCS) WRITE(6,490) TBCXP,TBCHGP,TBCTGP,TBCQGP,TBCPGP	NINPRTT 2276
488	FORMAT(1H+,65X,F6.2,2F8.1,F12.1,F8.1)	NINPRTT 2277
490	FORMAT(66X,F6.2,2F8.1,F12.1,F8.1)	NINPRTT 2278
499	CONTINUE	NINPRTT 2279
		NINPRTT 2280

94	CONTINUE	NINPRTT 2281
	DO 95 I = 1,200	NINPRTT 2282
95	NFLUID(I) = 0	NINPRTT 2283
100	WRITE(6,150) ICHNL	NINPRTT 2284
	IF (ICHNL.GT.1) GO TO 101	NINPRTT 2285
	IF (IHUB.EQ.1) WRITE(6,142)	NINPRTT 2286
	IF (IHUB.EQ.2) WRITE(6,144)	NINPRTT 2287
	IF (IHUB.EQ.3) WRITE(6,146)	NINPRTT 2288
101	CONTINUE	NINPRTT 2289
	IF (ICHNL.LT.NSLICE) GO TO 102	NINPRTT 2290
	IF (ITIP.EQ.1) WRITE(6,147)	NINPRTT 2291
	IF (ITIP.EQ.2) WRITE(6,148)	NINPRTT 2292
	IF (ITIP.EQ.3) WRITE(6,149)	NINPRTT 2293
102	CONTINUE	NINPRTT 2294
	TRIN = RIN(ICHNL)/CINCH(IUNITS)	NINPRTT 2295
	TROUT = ROUT(ICHNL)/CINCH(IUNITS)	NINPRTT 2296
	TDPLN = DPLN(ICHNL)/CINCH(IUNITS)	NINPRTT 2297
	TAPLN = APLN(ICHNL)/(CINCH(IUNITS)*CINCH(IUNITS))	NINPRTT 2298
	WRITE(6,103) TRIN, UL(IUNITS),TROUT,UL(IUNITS),	NINPRTT 2299
Z	TDPLN,UL(IUNITS),TAPLN,UA(IUNITS)	NINPRTT 2300
103	FORMAT(/' COOLANT PLENUM: RI=',F7.3,A4,' RO=',F7.3,A4,4X,	NINPRTT 2301
Z	' DHYD=',F7.4,A4,' APLEN=',F7.4,A4,'*2')	NINPRTT 2302
C		NINPRTT 2303
	IF (IUNITS.EQ.1) GO TO 500	NINPRTT 2304
C		NINPRTT 2305
	WRITE(6,153) NFWD,NSTA,SPAN	NINPRTT 2306
	WRITE(6,155) CD,ADUMP	NINPRTT 2307
	TEM = TTIO(1) + 460.	NINPRTT 2308
	WRITE(6,157) TEM,PTIO(1),PEX(ICHNL),WPLEN	NINPRTT 2309
	ITRBG = NFWD + 2	NINPRTT 2310
	WRITE(6,154) ICHNL,ITRBG	NINPRTT 2311
	DO 118 I = 1,NSTA,20	NINPRTT 2312
	IP18 = I + 18	NINPRTT 2313
	IF (IP18.GT.NSTA) IP18 = NSTA	NINPRTT 2314
	IF (I.EQ.1) WRITE(6,156) (J,J=I,IP18,2)	NINPRTT 2315
	IF (I.GT.1) WRITE(6,159) (J,J=I,IP18,2)	NINPRTT 2316
	ID = 0	NINPRTT 2317
	DO 104 J = I,IP18,2	NINPRTT 2318
	ID = ID + 1	NINPRTT 2319
	DUM1(ID) = RR(J)	NINPRTT 2320
	NFLUID(J) = 5*J	NINPRTT 2321
104	CONTINUE	NINPRTT 2322
	WRITE(6,158) (NFLUID(J),J=I,IP18,2)	NINPRTT 2323
	ID = 0	NINPRTT 2324
	DO 116 J = I,IP18,2	NINPRTT 2325
	ID = ID + 1	NINPRTT 2326
	NOS = NFLUID(J) - 4	NINPRTT 2327
	IF (NOS.GT.1) GO TO 106	NINPRTT 2328
	XOS = 0.0	NINPRTT 2329
	XJN = 0.0	NINPRTT 2330
	XMM = 0.0	NINPRTT 2331
	XIS = 0.0	NINPRTT 2332
	XCC = 0.0	NINPRTT 2333
	GO TO 108	NINPRTT 2334
106	XOS = XOS + DLX(NOS)	NINPRTT 2335
	XJN = XJN + DLX(NOS+1)	NINPRTT 2336
	XMM = XMM + DLX(NOS+2)	NINPRTT 2337
	XIS = XIS + DLX(NOS+3)	NINPRTT 2338
	XCC = XCC + DLX(NOS+4)	NINPRTT 2339
108	CONTINUE	NINPRTT 2340

DUM2 (ID) = XOS	NINPRTT 2341
DUM25 (ID) = XJN	NINPFTT 2342
DUM3 (ID) = XMM	NINPFTT 2343
DUM4 (ID) = XIS	NINPRTT 2344
DUM5 (ID) = XCC	NINPRTT 2345
DUM55 (ID) = TAU (NOS)	NINPRTT 2346
DUM6 (ID) = TAU (NOS+2)	NINPRTT 2347
NOS = NPLUID (J)	NINPRTT 2348
DUM7 (ID) = TAU (NOS)	NINPRTT 2349
DUM8 (ID) = A (NOS)	NINPRTT 2350
DUM9 (ID) = DH (J)	NINPRTT 2351
DUM10 (ID) = DHJ (J)	NINPRTT 2352
DUM11 (ID) = CNUM (J)	NINPRTT 2353
DUM12 (ID) = AJET (J)	NINPRTT 2354
DUM16 (ID) = THUBIN (J) - 460.	NINPRTT 2355
DUM17 (ID) = QHUBIN (J)	NINPRTT 2356
DUM18 (ID) = TTIPIN (J) - 460.	NINPRTT 2357
DUM19 (ID) = QTIPIN (J)	NINPRTT 2358
NOS = NPLUID (J) - 4	NINPRTT 2359
DUM13 (ID) = TG (J) - 460.	NINPRTT 2360
DUM14 (ID) = HG (J)	NINPRTT 2361
JHCAL = IHC (J)	NINPRTT 2362
DUM15 (ID) = HCAL (JHCAL)	NINPRTT 2363
IF (BTA.GT..01) DUM14 (ID) = QG (J)	NINPRTT 2364
116 CONTINUE	NINPRTT 2365
WRITE (6,160) (DUM1 (J),J=1,ID)	NINPRTT 2366
WRITE (6,162) (DUM2 (J),J=1,ID)	NINPFTT 2367
WRITE (6,163) (DUM25 (J),J=1,ID)	NINPRTT 2368
WRITE (6,164) (DUM3 (J),J=1,ID)	NINPFTT 2369
WRITE (6,166) (DUM4 (J),J=1,ID)	NINPFTT 2370
WRITE (6,168) (DUM5 (J),J=1,ID)	NINPFTT 2371
WRITE (6,169) (DUM55 (J),J=1,ID)	NINPRTT 2372
WRITE (6,170) (DUM6 (J),J=1,ID)	NINPRTT 2373
WRITE (6,172) (DUM7 (J),J=1,ID)	NINPRTT 2374
WRITE (6,174) (DUM8 (J),J=1,ID)	NINPRTT 2375
WRITE (6,176) (DUM9 (J),J=1,ID)	NINPRTT 2376
WRITE (6,178) (DUM10 (J),J=1,ID)	NINPRTT 2377
WRITE (6,180) (DUM11 (J),J=1,ID)	NINPFTT 2378
WRITE (6,182) (DUM12 (J),J=1,ID)	NINPRTT 2379
WRITE (6,183) (DUM15 (J),J=1,ID)	NINPRTT 2380
WRITE (6,184) (DUM13 (J),J=1,ID)	NINPRTT 2381
IF (BTA.LT..01) WRITE (6,186) (DUM14 (J),J=1,ID)	NINPRTT 2382
IF (BTA.GT..01) WRITE (6,188) (DUM14 (J),J=1,ID)	NINPRTT 2383
IF (ICHNL.GT.1) GO TO 118	NINPRTT 2384
IF (IHUB.EQ.1) WRITE (6,196) (DUM16 (J),J=1,ID)	NINPRTT 2385
IF (IHUB.EQ.3) WRITE (6,198) (DUM17 (J),J=1,ID)	NINPRTT 2386
IF (ITIP.EQ.1) WRITE (6,202) (DUM18 (J),J=1,ID)	NINPRTT 2387
IF (ITIP.EQ.3) WRITE (6,204) (DUM19 (J),J=1,ID)	NINPRTT 2388
118 CONTINUE	NINPFTT 2389
ITRBG = NFWD + 1	NINPFTT 2390
WRITE (6,190) ICHNL,ITRBG	NINPRTT 2391
XOS = 0.0	NINPFTT 2392
XJN = 0.0	NINPFTT 2393
XMM = 0.0	NINPRTT 2394
XIS = 0.0	NINPFTT 2395
XCC = 0.0	NINPRTT 2396
PO 140 I = 2,NSTA,20	NINPRTT 2397
IP18 = I + 18	NINPRTT 2398
IF (IP18.GT.NSTA) IP18 = NSTA-1	NINPRTT 2399
IF (I.EQ.2) WRITE (6,156) (J,J=1,IP18,2)	NINPRTT 2400

IF (I.GT.2) WRITE(6,159) (J,J=I,IP18,2)	NINPRTT 2401
ID = 0	NINPRTT 2402
DO 122 J = I,IP18,2	NINPRTT 2403
ID = ID + 1	NINPRTT 2404
DUM1(ID) = RR(J)	NINPRTT 2405
NFLUID(J) = 5*J	NINPRTT 2406
122 CONTINUE	NINPRTT 2407
WRITE(6,158) (NFLUID(J),J=I,IP18,2)	NINPRTT 2408
ID = 0	NINPRTT 2409
DO 130 J = I,IP18,2	NINPRTT 2410
ID = ID + 1	NINPRTT 2411
NOS = NFLUID(J) - 4	NINPRTT 2412
XOS = XOS + DLX(NOS)	NINPRTT 2413
XJN = XJN + DLX(NOS+1)	NINPRTT 2414
YMM = YMM + DLX(NOS+2)	NINPRTT 2415
XIS = XIS + DLX(NOS+3)	NINPRTT 2416
XCC = XCC + DLX(NOS+4)	NINPRTT 2417
DUM2(ID) = XOS	NINPRTT 2418
DUM25(ID) = XJN	NINPRTT 2419
DUM3(ID) = YMM	NINPRTT 2420
DUM4(ID) = XIS	NINPRTT 2421
DUM5(ID) = XCC	NINPRTT 2422
DUM55(ID) = TAU(NOS)	NINPRTT 2423
DUM6(ID) = TAU(NOS+2)	NINPRTT 2424
NOS = NFLUID(J)	NINPRTT 2425
DUM7(ID) = TAU(NOS)	NINPRTT 2426
DUM8(ID) = A(NOS)	NINPRTT 2427
DUM9(ID) = DH(J)	NINPRTT 2428
DUM10(ID) = DHJ(J)	NINPRTT 2429
DUM11(ID) = CNUM(J)	NINPRTT 2430
DUM12(ID) = AJET(J)	NINPRTT 2431
DUM16(ID) = THUBIN(J) - 460.	NINPRTT 2432
DUM17(ID) = QHUBIN(J)	NINPRTT 2433
DUM18(ID) = TTIPIN(J) - 460.	NINPRTT 2434
DUM19(ID) = QTIPIN(J)	NINPRTT 2435
NOS = NFLUID(J) - 4	NINPRTT 2436
DUM13(ID) = TG(J) - 460.	NINPRTT 2437
DUM14(ID) = HG(J)	NINPRTT 2438
JHCAL = IHC(J)	NINPRTT 2439
DUM15(ID) = HCAL(JHCAL)	NINPRTT 2440
IF (BTA.GT..01) DUM14(ID) = QG(J)	NINPRTT 2441
130 CONTINUE	NINPRTT 2442
WRITE(6,160) (DUM1(J),J=1,ID)	NINPRTT 2443
WRITE(6,162) (DUM2(J),J=1,ID)	NINPRTT 2444
WRITE(6,163) (DUM25(J),J=1,ID)	NINPRTT 2445
WRITE(6,164) (DUM3(J),J=1,ID)	NINPRTT 2446
WRITE(6,166) (DUM4(J),J=1,ID)	NINPRTT 2447
WRITE(6,168) (DUM5(J),J=1,ID)	NINPRTT 2448
WRITE(6,169) (DUM55(J),J=1,ID)	NINPRTT 2449
WRITE(6,170) (DUM6(J),J=1,ID)	NINPRTT 2450
WRITE(6,172) (DUM7(J),J=1,ID)	NINPRTT 2451
WRITE(6,174) (DUM8(J),J=1,ID)	NINPRTT 2452
WRITE(6,176) (DUM9(J),J=1,ID)	NINPRTT 2453
WRITE(6,178) (DUM10(J),J=1,ID)	NINPRTT 2454
WRITE(6,180) (DUM11(J),J=1,ID)	NINPRTT 2455
WRITE(6,182) (DUM12(J),J=1,ID)	NINPRTT 2456
WRITE(6,183) (DUM15(J),J=1,ID)	NINPRTT 2457
WRITE(6,184) (DUM13(J),J=1,ID)	NINPRTT 2458
IF (BTA.LT..01) WRITE(6,186) (DUM14(J),J=1,ID)	NINPRTT 2459
IF (BTA.GT..01) WRITE(6,188) (DUM14(J),J=1,ID)	NINPRTT 2460

	IF (ICHL.GT.1) GO TO 140	NINPRTT 2461
	IF (IHUB.EQ.1) WRITE(6,196) (DUM16(J),J=1,ID)	NINPRTT 2462
	IF (IHUB.EQ.3) WRITE(6,198) (DUM17(J),J=1,ID)	NINPRTT 2463
	IF (ITIP.EQ.1) WRITE(6,202) (DUM18(J),J=1,ID)	NINPRTT 2464
	IF (ITIP.EQ.3) WRITE(6,204) (DUM19(J),J=1,ID)	NINPRTT 2465
140	CONTINUE	NINPRTT 2466
150	FORMAT(1H1,46X,'INPUT FOR SLICE NUMBER',I3)	NINPRTT 2467
142	FORMAT(21X,' HUB TEMPERATURES ARE SPECIFIED')	NINPRTT 2468
144	FORMAT(21X,' ADIABATIC HUB SPECIFIED')	NINPRTT 2469
146	FORMAT(21X,' HUB HEAT FLUX IS SPECIFIED')	NINPRTT 2470
147	FORMAT(21X,' TIP TEMPERATURES ARE SPECIFIED')	NINPRTT 2471
148	FORMAT(21X,' ADIABATIC TIP SPECIFIED')	NINPRTT 2472
149	FORMAT(21X,' TIP HEAT FLUX IS SPECIFIED')	NINPRTT 2473
153	FORMAT(/' NUMBER OF STATIONS IN IMPINGEMENT REGION IS',I3,	NINPRTT 2474
Z	' , TOTAL NUMBER OF STATIONS IS',I3,	NINPRTT 2475
Z	' , SPAN OF THIS SLICE IS',F6.3,' IN')	NINPRTT 2476
155	FORMAT(' IMPINGEMENT HOLE DISCHARGE COEF.=' ,F6.3,	NINPRTT 2477
Z	' , AREA OF DUMP TO TRAILING EDGE =' ,F8.5,' IN**2')	NINPRTT 2478
157	FORMAT(' COOLANT INLET TEMP.=' ,F7.1,' R, COOLANT INLET PRESSURE',	NINPRTT 2479
Z	' =' ,F6.1,' PSIA, EXIT PRESSURE =' ,	NINPRTT 2480
Z	F6.1,' PSIA,/' COOLANT FLOW =' ,F6.1,' LBM/HR')	NINPRTT 2481
154	FORMAT(/' PRESSURE SIDE, SLICE ',I2,' , TRAILING EDGE REGION ',	NINPRTT 2482
Z	' BEGINS AT STATION-',I3)	NINPRTT 2483
156	FORMAT(/' STATION NUMBER',5X,10(6X,I4))	NINPRTT 2484
158	FORMAT(' COOLANT NODE NUMBER',10(6X,I4))	NINPRTT 2485
159	FORMAT(1H2//' STATION NUMBER',5X,10(6X,I4))	NINPRTT 2486
160	FORMAT(' RADIAL LOCATION (IN)',10F10.3)	NINPRTT 2487
162	FORMAT(' X, OUTSIDE SUR. (IN)',10F10.5)	NINPRTT 2488
163	FORMAT(' X, INTERFACE (IN)',10F10.5)	NINPRTT 2489
164	FORMAT(' X, MID-METAL (IN)',10F10.5)	NINPRTT 2490
166	FORMAT(' X, INSIDE SURF. (IN)',10F10.5)	NINPRTT 2491
168	FORMAT(' X, MID.COOL.CH. (IN)',10F10.5)	NINPRTT 2492
169	FORMAT(/' COATING THKNSS (IN)',10F10.5)	NINPRTT 2493
170	FORMAT(' WALL THICKNESS (IN)',10F10.5)	NINPRTT 2494
172	FORMAT(' CHANNEL WIDTH (IN)',10F10.5)	NINPRTT 2495
174	FORMAT(' CHANNEL AREA(IN**2)',10F10.5)	NINPRTT 2496
176	FORMAT(' CHANNEL HYD.DIA(IN)',10F10.5)	NINPRTT 2497
178	FORMAT(/' IMP.JET HYD.DIA(IN)',10F10.5)	NINPRTT 2498
180	FORMAT(' NO. OF IMP. JETS ',10F10.2)	NINPRTT 2499
182	FORMAT(' TOT.JET AREA(IN**2)',10F10.5)	NINPRTT 2500
183	FORMAT(/' TYPE OF HC CALC. ',10(6X,A4))	NINPRTT 2501
184	FORMAT(' OUTSIDE BC: TG,(F)',10F10.1)	NINPRTT 2502
186	FORMAT(' HG (BTU/HR/FT**2/R)',10F10.1)	NINPRTT 2503
188	FORMAT(' QG (BTU/HR/FT**2)',10F10.1)	NINPRTT 2504
190	FORMAT(1H1,' SUCTION SIDE, SLICE ',I2,' , TRAILING EDGE REGION',	NINPRTT 2505
Z	' BEGINS AT STATION-',I3)	NINPRTT 2506
192	FORMAT(/' CLAD K(BTU/HR/FT/R)',10F10.3)	NINPRTT 2507
194	FORMAT(' METL K(BTU/HR/FT/R)',10F10.3)	NINPRTT 2508
196	FORMAT(' GIVEN HUB TEMP. (F)',10F10.1)	NINPRTT 2509
198	FORMAT(' QHUB (BTU/HR/FT**2)',10F10.1)	NINPRTT 2510
202	FORMAT(' GIVEN TIP TEMP. (F)',10F10.1)	NINPRTT 2511
204	FORMAT(' QTIP (BTU/HR/FT**2)',10F10.1)	NINPRTT 2512
C		NINPRTT 2513
	GO TO 350	NINPRTT 2514
C		NINPRTT 2515
500	SPANC = SPAN/CINCH(1)	NINPRTT 2516
	ADUMPC = ADUMP/(CINCH(1)**2)	NINPRTT 2517
	WRITE(6,553) NFWD,NSPA,SPANC	NINPRTT 2518
	WRITE(6,555) CD,ADUMPC	NINPRTT 2519
	TEM = (TTIO(1) + 460.)/1.8	NINPRTT 2520

CONTENTS

	Page	
SUMMARY	1	1/A5
INTRODUCTION	1	1/A5
OVERVIEW	2	1/A6
Method of Analysis	2	1/A6
Blade geometric model	2	1/A6
Numerical model	3	1/A7
General Program Description	3	1/A7
DETAILED PROGRAM PROCEDURE	5	1/A9
Plock Data NGASDAT	6	1/A10
Main Program NTTACT	6	1/A10
Subroutine FLOWS	7	1/A11
Subroutine FLSPLT	7	1/A11
Subroutine GASTBL	8	1/A12
Subroutine GAUSS	8	1/A12
Subroutine GETIN	8	1/A12
Function HCFRCD	9	1/A13
Subroutine HCOOL	9	1/A13
Subroutine HCPINS	9	1/A13
Subroutine INPRT	9	1/A13
Subroutine PARRAY	9	1/A13
Subroutine PLNUM	10	1/A14
Subroutine PLOTMF	10	1/A14
Subroutine PREP	10	1/A14
Subroutine TARRAY	10	1/A14
Subroutine TCOEF	11	1/B1
Subroutine THRCON	11	1/B1
Subroutine WROUT	11	1/B1
DICTIONARY	11	1/B1
PROGRAM LISTING	53	1/E1
REFERENCES	150	2/E1

	PTIOC = PTIO(1)/CPRSR(1)	NINPRTT 2521
	PEXC = PEX(ICHNL)/CPRSR(1)	NINPRTT 2522
	WPLENC = WPLEN/CMSPL(1)	NINPRTT 2523
	WRITE(6,557) TEM,PTIOC,PEXC,WPLENC	NINPRTT 2524
	ITRBG = NFWD + 2	NINPRTT 2525
	WRITE(6,154) ICHNL,ITRBG	NINPRTT 2526
	DO 518 I = 1,NSTA,20	NINPRTT 2527
	IP18 = I + 18	NINPRTT 2528
	IF (IP18.GT.NSTA) IP18 = NSTA	NINPRTT 2529
	IF (I.EQ.1) WRITE(6,556) (J,J=I,IP18,2)	NINPRTT 2530
	IF (I.GT.1) WRITE(6,559) (J,J=I,IP18,2)	NINPRTT 2531
	ID = 0	NINPRTT 2532
	DO 504 J = I,IP18,2	NINPRTT 2533
	ID = ID + 1	NINPRTT 2534
	DUM1(ID) = RR(J)/CINCH(1)	NINPRTT 2535
	NFLUID(J) = 5*J	NINPRTT 2536
504	CONTINUE	NINPRTT 2537
	WRITE(6,558) (NFLUID(J),J=I,IP18,2)	NINPRTT 2538
	ID = 0	NINPRTT 2539
	DO 516 J = I,IP18,2	NINPRTT 2540
	ID = ID + 1	NINPRTT 2541
	NOS = NFLUID(J) - 4	NINPRTT 2542
	IF (NOS.GT.1) GO TO 506	NINPRTT 2543
	XOS = 0.0	NINPRTT 2544
	XJN = 0.0	NINPRTT 2545
	XMM = 0.0	NINPRTT 2546
	XIS = 0.0	NINPRTT 2547
	XCC = 0.0	NINPRTT 2548
	GO TO 508	NINPRTT 2549
506	XOS = XOS + DLX(NOS)/CINCH(1)	NINPRTT 2550
	XJN = XJN + DLX(NOS+1)/CINCH(1)	NINPRTT 2551
	XMM = XMM + DLX(NOS+2)/CINCH(1)	NINPRTT 2552
	XIS = XIS + DLX(NOS+3)/CINCH(1)	NINPRTT 2553
	XCC = XCC + DLX(NOS+4)/CINCH(1)	NINPRTT 2554
508	CONTINUE	NINPRTT 2555
	DUM2(ID) = XOS	NINPRTT 2556
	DUM25(ID) = XJN	NINPRTT 2557
	DUM3(ID) = XMM	NINPRTT 2558
	DUM4(ID) = XIS	NINPRTT 2559
	DUM5(ID) = XCC	NINPRTT 2560
	DUM55(ID) = TAU(NOS)/CINCH(1)	NINPRTT 2561
	DUM6(ID) = TAU(NOS+2)/CINCH(1)	NINPRTT 2562
	NOS = NFLUID(J)	NINPRTT 2563
	DUM7(ID) = TAU(NOS)/CINCH(1)	NINPRTT 2564
	DUM8(ID) = A(NOS)/(CINCH(1)**2)	NINPRTT 2565
	DUM9(ID) = DH(J)/CINCH(1)	NINPRTT 2566
	DUM10(ID) = DHJ(J)/CINCH(1)	NINPRTT 2567
	DUM11(ID) = CNUM(J)	NINPRTT 2568
	DUM12(ID) = AJET(J)/(CINCH(1)**2)	NINPRTT 2569
	DUM16(ID) = THUBIN(J)/1.8	NINPRTT 2570
	DUM17(ID) = QHUBIN(J)/CHFLX(1)	NINPRTT 2571
	DUM18(ID) = TTIPIN(J)/1.8	NINPRTT 2572
	DUM19(ID) = QTIPIN(J)/CHFLX(1)	NINPRTT 2573
	NOS = NFLUID(J) - 4	NINPRTT 2574
	DUM13(ID) = TG(J)/1.8	NINPRTT 2575
	DUM14(ID) = HG(J)/CHTC(1)	NINPRTT 2576
	JHCAL = IHC(J)	NINPRTT 2577
	DUM15(ID) = HCAL(JHCAL)	NINPRTT 2578
	IF (BTA.GT..01) DUM14(ID) = QG(J)/CHFLX(1)	NINPRTT 2579
516	CONTINUE	NINPRTT 2580

WRITE(6,560)	(DUM1(J),J=1,ID)	NINPRTT	2581
WRITE(6,562)	(DUM2(J),J=1,ID)	NINPRTT	2582
WRITE(6,563)	(DUM25(J),J=1,ID)	NINPRTT	2583
WRITE(6,564)	(DUM3(J),J=1,ID)	NINPRTT	2584
WRITE(6,566)	(DUM4(J),J=1,ID)	NINPRTT	2585
WRITE(6,568)	(DUM5(J),J=1,ID)	NINPRTT	2586
WRITE(6,569)	(DUM55(J),J=1,ID)	NINPRTT	2587
WRITE(6,570)	(DUM6(J),J=1,ID)	NINPRTT	2588
WRITE(6,572)	(DUM7(J),J=1,ID)	NINPRTT	2589
WRITE(6,574)	(DUM8(J),J=1,ID)	NINPRTT	2590
WRITE(6,576)	(DUM9(J),J=1,ID)	NINPRTT	2591
WRITE(6,578)	(DUM10(J),J=1,ID)	NINPRTT	2592
WRITE(6,580)	(DUM11(J),J=1,ID)	NINPRTT	2593
WRITE(6,582)	(DUM12(J),J=1,ID)	NINPRTT	2594
WRITE(6,583)	(DUM15(J),J=1,ID)	NINPRTT	2595
WRITE(6,584)	(DUM13(J),J=1,ID)	NINPRTT	2596
IF (BTA.LT..01)	WRITE(6,586) (DUM14(J),J=1,ID)	NINPRTT	2597
IF (BTA.GT..01)	WRITE(6,588) (DUM14(J),J=1,ID)	NINPRTT	2598
IF (ICHNL.GT.1)	GO TO 518	NINPRTT	2599
IF (IHUB.EQ.1)	WRITE(6,596) (DUM16(J),J=1,ID)	NINPRTT	2600
IF (IHUB.EQ.3)	WRITE(6,598) (DUM17(J),J=1,ID)	NINPRTT	2601
IF (ITIP.EQ.1)	WRITE(6,602) (DUM18(J),J=1,ID)	NINPRTT	2602
IF (ITIP.EQ.3)	WRITE(6,604) (DUM19(J),J=1,ID)	NINPRTT	2603
518 CONTINUE		NINPRTT	2604
ITRBG = NFWD + 1		NINPRTT	2605
WRITE(6,190)	ICHNL,ITRBG	NINPRTT	2606
XOS = 0.0		NINPRTT	2607
XJN = 0.0		NINPRTT	2608
XMM = 0.0		NINPRTT	2609
XIS = 0.0		NINPRTT	2610
XCC = 0.0		NINPRTT	2611
DO 540 I = 2,NSTA,20		NINPRTT	2612
IP18 = I + 18		NINPRTT	2613
IF (IP18.GT.NSTA)	IP18 = NSTA-1	NINPRTT	2614
IF (I.EQ.2)	WRITE(6,556) (J,J=I,IP18,2)	NINPRTT	2615
IF (I.GT.2)	WRITE(6,559) (J,J=I,IP18,2)	NINPRTT	2616
ID = 0		NINPRTT	2617
DO 522 J = I,IP18,2		NINPRTT	2618
ID = ID + 1		NINPRTT	2619
DUM1(ID) = RR(J)/CINCH(1)		NINPRTT	2620
NFLUID(J) = 5*J		NINPRTT	2621
522 CONTINUE		NINPRTT	2622
WRITE(6,558)	(NFLUID(J),J=I,IP18,2)	NINPRTT	2623
ID = 0		NINPRTT	2624
DO 530 J = I,IP18,2		NINPRTT	2625
ID = ID + 1		NINPRTT	2626
NOS = NFLUID(J) - 4		NINPRTT	2627
XOS = XOS + DLX(NOS)/CINCH(1)		NINPRTT	2628
XJN = XJN + DLX(NOS+1)/CINCH(1)		NINPRTT	2629
XMM = XMM + DLX(NOS+2)/CINCH(1)		NINPRTT	2630
XIS = XIS + DLX(NOS+3)/CINCH(1)		NINPRTT	2631
XCC = XCC + DLX(NOS+4)/CINCH(1)		NINPRTT	2632
DUM2(ID) = XOS		NINPRTT	2633
DUM25(ID) = XJN		NINPRTT	2634
DUM3(ID) = XMM		NINPRTT	2635
DUM4(ID) = XIS		NINPRTT	2636
DUM5(ID) = XCC		NINPRTT	2637
DUM55(ID) = TAU(NOS)/CINCH(1)		NINPRTT	2638
DUM6(ID) = TAU(NOS+2)/CINCH(1)		NINPRTT	2639
NOS = NFLUID(J)		NINPRTT	2640

DUM7 (ID) = TAU (NOS) / CINCH (1)	NINPRTT 2641
DUM8 (ID) = A (NOS) / (CINCH (1) **2)	NINPRTT 2642
DUM9 (ID) = DH (J) / CINCH (1)	NINPRTT 2643
DUM10 (ID) = DHJ (J) / CINCH (1)	NINPRTT 2644
DUM11 (ID) = CNUM (J)	NINPRTT 2645
DUM12 (ID) = AJET (J) / (CINCH (1) **2)	NINPRTT 2646
DUM16 (ID) = THUBIN (J) / 1.8	NINPRTT 2647
DUM17 (ID) = QHUBIN (J) / CHFLX (1)	NINPRTT 2648
DUM18 (ID) = TTIPIN (J) / 1.8	NINPRTT 2649
DUM19 (ID) = QTIPIN (J) / CHFLX (1)	NINPRTT 2650
NOS = NFLUID (J) - 4	NINPRTT 2651
DUM13 (ID) = TG (J) / 1.8	NINPRTT 2652
DUM14 (ID) = HG (J) / CHTC (1)	NINPRTT 2653
JHCAL = IHC (J)	NINPRTT 2654
DUM15 (ID) = HCAL (JHCAL)	NINPRTT 2655
IF (BTA.GT..01) DUM14 (ID) = QG (J) / CHFLX (1)	NINPRTT 2656
530 CONTINUE	NINPRTT 2657
WRITE (6,560) (DUM1 (J), J=1, ID)	NINPRTT 2658
WRITE (6,562) (DUM2 (J), J=1, ID)	NINPRTT 2659
WRITE (6,563) (DUM25 (J), J=1, ID)	NINPRTT 2660
WRITE (6,564) (DUM3 (J), J=1, ID)	NINPRTT 2661
WRITE (6,566) (DUM4 (J), J=1, ID)	NINPRTT 2662
WRITE (6,568) (DUM5 (J), J=1, ID)	NINPRTT 2663
WRITE (6,569) (DUM55 (J), J=1, ID)	NINPRTT 2664
WRITE (6,570) (DUM6 (J), J=1, ID)	NINPRTT 2665
WRITE (6,572) (DUM7 (J), J=1, ID)	NINPRTT 2666
WRITE (6,574) (DUM8 (J), J=1, ID)	NINPRTT 2667
WRITE (6,576) (DUM9 (J), J=1, ID)	NINPRTT 2668
WRITE (6,578) (DUM10 (J), J=1, ID)	NINPRTT 2669
WRITE (6,580) (DUM11 (J), J=1, ID)	NINPRTT 2670
WRITE (6,582) (DUM12 (J), J=1, ID)	NINPRTT 2671
WRITE (6,583) (DUM15 (J), J=1, ID)	NINPRTT 2672
WRITE (6,584) (DUM13 (J), J=1, ID)	NINPRTT 2673
IF (BTA.LT..01) WRITE (6,586) (DUM14 (J), J=1, ID)	NINPRTT 2674
IF (BTA.GT..01) WRITE (6,588) (DUM14 (J), J=1, ID)	NINPRTT 2675
IF (ICHNL.GT.1) GO TO 540	NINPRTT 2676
IF (IHUB.EQ.1) WRITE (6,596) (DUM16 (J), J=1, ID)	NINPRTT 2677
IF (IHUB.EQ.3) WRITE (6,598) (DUM17 (J), J=1, ID)	NINPRTT 2678
IF (ITIP.EQ.1) WRITE (6,602) (DUM18 (J), J=1, ID)	NINPRTT 2679
IF (ITIP.EQ.3) WRITE (6,604) (DUM19 (J), J=1, ID)	NINPRTT 2680
540 CONTINUE	NINPRTT 2681
553 FORMAT (/ ' NUMBER OF STATIONS IN IMPINGEMENT REGION IS', I3,	NINPRTT 2682
Z ' , TOTAL NUMBER OF STATIONS IS', I3,	NINPRTT 2683
Z ' , SPAN OF THIS SLICE IS', F6.3, ' CM')	NINPRTT 2684
555 FORMAT (' IMPINGEMENT HOLE DISCHARGE COEF.=', F6.3,	NINPRTT 2685
Z ' , AREA OF DUMP TO TRAILING EDGE =', F8.5, ' CM**2')	NINPRTT 2686
557 FORMAT (' COOLANT INLET TEMP.=', F7.1, ' K, COOLANT INLET',	NINPRTT 2687
Z ' PRESSURE =', F7.1, ' KPA, EXIT PRESSURE =',	NINPRTT 2688
Z F7.1, ' KPA, ' / ' COOLANT FLOW =', F6.1, ' KG/HR')	NINPRTT 2689
556 FORMAT (/ ' STATION NUMBER', 5X, 10 (6X, I4))	NINPRTT 2690
558 FORMAT (' COOLANT NODE NUMBER', 10 (6X, I4))	NINPRTT 2691
559 FORMAT (1H2 / ' STATION NUMBER', 5X, 10 (6X, I4))	NINPRTT 2692
560 FORMAT (' RADIAL LOCATION (CM)', 10F10.3)	NINPRTT 2693
562 FORMAT (' X, OUTSIDE SUR. (CM)', 10F10.5)	NINPRTT 2694
563 FORMAT (' X, INTERFACE (CM)', 10F10.5)	NINPRTT 2695
564 FORMAT (' X, MID-METAL (CM)', 10F10.5)	NINPRTT 2696
566 FORMAT (' X, INSIDE SURF. (CM)', 10F10.5)	NINPRTT 2697
568 FORMAT (' X, MID.COOL.CH. (CM)', 10F10.5)	NINPRTT 2698
569 FORMAT (/ ' COATING THKNSS (CM)', 10F10.5)	NINPRTT 2699
570 FORMAT (' WALL THICKNESS (CM)', 10F10.5)	NINPRTT 2700

572	FORMAT(' CHANNEL WIDTH (CM) ',10F10.5)	NINPRTT 2701
574	FORMAT(' CHANNEL AREA(CM**2) ',10F10.5)	NINPRTT 2702
576	FORMAT(' CHANNEL HYD.DIA(CM) ',10F10.5)	NINPRTT 2703
578	FORMAT('/ IMP.JET HYD.DIA(CM) ',10F10.5)	NINPRTT 2704
580	FORMAT(' NO. OF IMP. JETS ',10F10.2)	NINPRTT 2705
582	FORMAT(' TOT.JET AREA(CM**2) ',10F10.5)	NINPRTT 2706
583	FORMAT('/ TYPE OF HC CALC. ',10(6X,A4))	NINPRTT 2707
584	FORMAT(' OUTSIDE BC: TG,(K) ',10F10.1)	NINPRTT 2708
586	FORMAT(' HG (W/M**2/K) ',10F10.1)	NINPRTT 2709
588	FORMAT(' QG (W/M**2) ',10F10.1)	NINPRTT 2710
592	FORMAT(' CLAD K (W/M/K) ',10F10.3)	NINPRTT 2711
594	FORMAT(' METAL K (W/M/K) ',10F10.3)	NINPRTT 2712
596	FORMAT(' GIVEN HUB TEMP. (K) ',10F10.1)	NINPRTT 2713
598	FORMAT(' QHUB (W/M**2) ',10F10.1)	NINPRTT 2714
602	FORMAT(' GIVEN TIP TEMP. (K) ',10F10.1)	NINPRTT 2715
604	FORMAT(' QTIP (W/M**2) ',10F10.1)	NINPRTT 2716
C		NINPRTT 2717
350	CONTINUE	NINPRTT 2718
	RETURN	NINPRTT 2719
	END	NINPRTT 2720

C----	SOURCE.NPARAYT	NPARAYT 2721
	SUBROUTINE PARRAY(JS,JSSENS,ICHOKE)	NPARAYT 2722
C		NPARAYT 2723
C-	SOURCE.NPARAYT----	NPARAYT 2724
C+++	A SUBROUTINE TO SET UP THE COEF ARRAY TO SOLVE FOR BLADE PRESSURES	NPARAYT 2725
C		NPARAYT 2726
	REAL*8 TCOF	NPARAYT 2727
C		NPARAYT 2728
	COMMON /MATRX/ TCOF(400,30)	NPARAYT 2729
C		NPARAYT 2730
	COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),	NPARAYT 2731
Z	CPC(80), GAMC(80), DUMR1(80), DUMR2(80)	NPARAYT 2732
C		NPARAYT 2733
	COMMON /TCO/ ADUMP, BTA, CD, CP,	NPARAYT 2734
Z	GAM, PIM, R, SPAN, TOG,	NPARAYT 2735
Z	WDUMP, WIN, AKC(15,80), AKW(15,80),	NPARAYT 2736
Z	A(400), AJET(80), AM2(80), CNUM(80),	NPARAYT 2737
Z	DH(80), DHF(80), DHJ(80),	NPARAYT 2738
Z	DLX(400), PF(80), HC(80), HG(80),	NPARAYT 2739
Z	P(2,15,80), PEXIT(15), PUMP(80), QG(80),	NPARAYT 2740
Z	QSNK(80), RR(80), S(15), T(2,15,400),	NPARAYT 2741
Z	TG(80), TAU(400), WFC(80),	NPARAYT 2742
Z	WJ(15,80), WCROS(2,15,80), XN(80),	NPARAYT 2743
Z	ICOR, IFILM, IHUB, ITIP,	NPARAYT 2744
Z	ISBLOK, ISLICE, NBLKSZ, NSLICE,	NPARAYT 2745
Z	NPWD, NSTA, IHC(80)	NPARAYT 2746
C		NPARAYT 2747
	COMMON /TRANSNT/ RHOC, RHOM, SPHTC, SPHTM,	NPARAYT 2748
Z	DLTYNE, TYNE, TEPS, TYNMAX	NPARAYT 2749
C		NPARAYT 2750
	DIMENSION POLD(80), PSAV(5)	NPARAYT 2751
C		NPARAYT 2752
C	COMPUTE NEW PRESSURES	NPARAYT 2753
C		NPARAYT 2754
C	IPNL = THE NUMBER OF FLOW CHANNEL NODES	NPARAYT 2755
C		NPARAYT 2756
	TREPS = 1.0	NPARAYT 2757
	IF (TYNE.GE.0.) TREPS = TEPS	NPARAYT 2758
800	IPNL = NSTA - 3	NPARAYT 2759
	MODST = 5*NSTA	NPARAYT 2760

NODSP = 5*NPWD	NPARAYT 2761
C	NPARAYT 2762
C INITIALIZE COEFFICIENT ARRAY TO 0.0	NPARAYT 2763
C	NPARAYT 2764
DO 810 I = 1,IPNL	NPARAYT 2765
DO 810 J = 1,30	NPARAYT 2766
810 TCOF(I,J) = 0.0	NPARAYT 2767
C	NPARAYT 2768
C COMPUTE THE COEFFICIENT VALUES	NPARAYT 2769
C	NPARAYT 2770
DO 900 I = 1,IPNL	NPARAYT 2771
FILM = 0.0	NPARAYT 2772
820 ICHK = I - 2*(I/2)	NPARAYT 2773
C	NPARAYT 2774
C FOR THE IMPINGEMENT REGION:	NPARAYT 2775
C ICHK = 0 IMPLIES I IS EVEN AND STATION IS ON SUCTION SIDE	NPARAYT 2776
C = 1 IMPLIES THAT I IS ODD AND STATION IS ON PRESSURE SIDE	NPARAYT 2777
C	NPARAYT 2778
C DEFINE THE REAL NODE NUMBER IN TERMS OF I	NPARAYT 2779
C WHERE IPL IS THE PIVOTAL ELEMENT = COOLANT NODE NUMBER, LCOOL	NPARAYT 2780
C IDN = DOWNSTREAM COOLANT NODE	NPARAYT 2781
C IUP = UPSTREAM COOLANT NODE	NPARAYT 2782
C	NPARAYT 2783
IF (I.LT.NPWD) GO TO 840	NPARAYT 2784
IF (I.EQ.NPWD) GO TO 890	NPARAYT 2785
C	NPARAYT 2786
C FOR I=NPWD, THE NODE IS THE ENTRANCE TO THE TRAILING EDGE AND IS	NPARAYT 2787
C TREATED SEPARATELY AT (890)	NPARAYT 2788
C FOR I>NPWD, THE NODE IS IN THE TRAILING EDGE AND IRL IS DEFINED AS:	NPARAYT 2789
C	NPARAYT 2790
IF (ICHK.GT.0) GO TO 885	NPARAYT 2791
IRL = 5*I	NPARAYT 2792
IDN = IRL + 10	NPARAYT 2793
IDNS = I+2	NPARAYT 2794
IUP = IRL	NPARAYT 2795
IUPS = IDNS - 2	NPARAYT 2796
ITC = 10	NPARAYT 2797
ITCP = 12	NPARAYT 2798
830 CONTINUE	NPARAYT 2799
GO TO 860	NPARAYT 2800
C	NPARAYT 2801
840 CONTINUE	NPARAYT 2802
IRL = 5*I	NPARAYT 2803
IF (I.GT.JS) GO TO 843	NPARAYT 2804
C	NPARAYT 2805
IF (I.LT.JS) GO TO 852	NPARAYT 2806
C	NPARAYT 2807
IF (ICHK.GT.0) GO TO 849	NPARAYT 2808
C	NPARAYT 2809
GO TO 855	NPARAYT 2810
C	NPARAYT 2811
843 IF (ICHK.GT.0) GO TO 849	NPARAYT 2812
C	NPARAYT 2813
C STATION I IS SUCTION SIDE, DOWNSTREAM OF SPLIT POINT	NPARAYT 2814
C	NPARAYT 2815
846 IUPS = I - 2	NPARAYT 2816
IDNS = I	NPARAYT 2817
IUP = IRL - 10	NPARAYT 2818
IDN = IRL	NPARAYT 2819
ITC = 8	NPARAYT 2820

ITCP = 10	NPARAYT 2821
IF (I.GT.2) GO TO 860	NPARAYT 2822
IUPS = 1	NPARAYT 2823
IUP = 5	NPARAYT 2824
ITC = 9	NPARAYT 2825
GO TO 860	NPARAYT 2826
C	NPARAYT 2827
C STATION I IS PRESSURE SIDE, DOWNSTREAM OF SPLIT POINT	NPARAYT 2828
C	NPARAYT 2829
849 CONTINUE	NPARAYT 2830
IUPS = I	NPARAYT 2831
IDNS = I + 2	NPARAYT 2832
IUP = IRL	NPARAYT 2833
IDN = IRL + 10	NPARAYT 2834
ITC = 10	NPARAYT 2835
ITCP = 12	NPARAYT 2836
GO TO 860	NPARAYT 2837
852 CONTINUE	NPARAYT 2838
IF (ICLK.GT.0) GO TO 858	NPARAYT 2839
IF (ICLK.NE.JSENS) GO TO 846	NPARAYT 2840
855 CONTINUE	NPARAYT 2841
C	NPARAYT 2842
C I IS ON SUCTION SIDE, FORWARD OF SPLIT POINT	NPARAYT 2843
C	NPARAYT 2844
IUPS = I	NPARAYT 2845
IDNS = I - 2	NPARAYT 2846
IUP = IRL	NPARAYT 2847
IDN = IRL - 10	NPARAYT 2848
IDX = IUP	NPARAYT 2849
ITC = 10	NPARAYT 2850
ITCP = 8	NPARAYT 2851
IF (I.GT.2) GO TO 860	NPARAYT 2852
IDNS = 1	NPARAYT 2853
IDN = 5	NPARAYT 2854
ITCP = 9	NPARAYT 2855
GO TO 860	NPARAYT 2856
858 CONTINUE	NPARAYT 2857
IF (ICLK.NE.JSENS) GO TO 849	NPARAYT 2858
IDNS = I	NPARAYT 2859
IUPS = I + 2	NPARAYT 2860
IDN = IRL	NPARAYT 2861
IUP = IRL + 10	NPARAYT 2862
IDX = IUP	NPARAYT 2863
ITC = 12	NPARAYT 2864
ITCP = 10	NPARAYT 2865
C	NPARAYT 2866
860 CONTINUE	NPARAYT 2867
C	NPARAYT 2868
C	NPARAYT 2869
TRTRM = 0.0	NPARAYT 2870
IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRM = 12.*DLX(IDN) *	NPARAYT 2871
Z (WCROS(2,ISLICE,IDNS)-WCROS(1,ISLICE,IDNS))/(DLTYME*A(IUP)*32.2)	NPARAYT 2872
WPCDUM = WFC(IDNS)	NPARAYT 2873
IF (I.GT.NFWD) WPCDUM = WPCDUM + WFC(IDNS+1)	NPARAYT 2874
IF (WCROS(2,ISLICE,IDNS).NE.0.0) FILM = WPCDUM/WCROS(2,ISLICE,IDNS)	NPARAYT 2875
TCOP(I,ITC) = TREPS*	NPARAYT 2876
Z ((1.0 + GAMC(IUPS)*AM2(IUPS)) + (A(IDN)-A(IUP))/(2.*A(IUP)))	NPARAYT 2877
TCOP(I,ITCP) = TREPS*(-(1.0 + .5*GAMC(IDNS)*AM2(IDNS))*	NPARAYT 2878
Z (4.*FF(IDNS)*DLX(IDX)/DH(IDNS)+2.+2.*FILM))*A(IDN)/A(IUP)	NPARAYT 2879
Z + (A(IDN)-A(IUP))/(2.*A(IUP)))	NPARAYT 2880

ROOT	= SQRT(32.2*GAMC(IDNS)*R*T(2,ISLICE,IDN)*AM2(IDNS))	NPARAYT	2881
PUMTRM	= 0.0	NPARAYT	2882
IF (ROOT.NE.0.0)	PUMTRM = (3.14159265*WS/30.):**2*	NPARAYT	2883
Z	RR(IDNS)*(RR(IDNS)-RR(IUPS))*WCROS(2,ISLICE,IDNS)	NPARAYT	2884
Z	/(A(IUP)*ROOT*144.*32.2)	NPARAYT	2885
TCOF(I,20)	= -PUMTRM + TRTRM - (1.-TREPS)*	NPARAYT	2886
Z	(P(1,ISLICE,IUPS)*TCOF(I,ITC) + P(1,ISLICE,IDNS)*TCOF(I,ITCP))	NPARAYT	2887
370	CONTINUE	NPARAYT	2888
IF (IDNS.NE.ICHOKE)	GO TO 880	NPARAYT	2889
TCOF(I,20)	= -P(1,ISLICE,ICHOKE)*TCOF(I,12) + TCOF(I,20)	NPARAYT	2890
TCOF(I,12)	= 0.0	NPARAYT	2891
880	CONTINUE	NPARAYT	2892
C		NPARAYT	2893
C	FOR TRAILING EDGE CHANNELS:	NPARAYT	2894
C		NPARAYT	2895
IF (I.LT.IFNL)	GO TO 900	NPARAYT	2896
C		NPARAYT	2897
C	TCOF(I,20) IS NON-ZERO ONLY FOR I=IFNL	NPARAYT	2898
C		NPARAYT	2899
IF (ICHOKE.EQ.NSTA-1)	GO TO 900	NPARAYT	2900
TCOF(I,20)	= - PEXIT(ISLICE)*TCOF(I,12) + TCOF(I,20)	NPARAYT	2901
TCOF(I,12)	= 0.0	NPARAYT	2902
GO TO 900		NPARAYT	2903
885	CONTINUE	NPARAYT	2904
C		NPARAYT	2905
C	FOR A PRESSURE SIDE, TRAILING EDGE REGION STATION, COOLANT NODE	NPARAYT	2906
C	IS IDENTICAL TO SUCTION SIDE NODE.	NPARAYT	2907
C		NPARAYT	2908
TCOF(I,10)	= 1.0	NPARAYT	2909
TCOF(I,9)	= -1.0	NPARAYT	2910
TCOF(I,20)	= 0.0	NPARAYT	2911
GO TO 900		NPARAYT	2912
890	CONTINUE	NPARAYT	2913
C		NPARAYT	2914
C	FOR THE SPECIAL NODE AT THE ENTRANCE TO THE TRAILING EDGE:	NPARAYT	2915
C	ALLOWING FOR THE POSSIBILITY OF ADDITION OF EXTRA COOLING AIR	NPARAYT	2916
C	INTO TRAILING EDGE,	NPARAYT	2917
C		NPARAYT	2918
C		NPARAYT	2919
TRTRM	= 0.0	NPARAYT	2920
IF (DLTYME.GT.0.0.AND.TYME.GE.0.)	TRTRM=12.*DLX(NFWD+1)*	NPARAYT	2921
Z	(WCROS(2,ISLICE,NFWD+1)-WCROS(1,ISLICE,NFWD+1))/	NPARAYT	2922
Z	(DLTYME*A(NODSF+5)*32.2)	NPARAYT	2923
AVRGA	= (A(NODSF-5) + A(NODSF) - A(NODSF+5))/(3.*A(NODSF+5))	NPARAYT	2924
TCOF(I,9)	= TREPS*((1. + GAMC(NFWD-1)*AM2(NFWD-1))*	NPARAYT	2925
Z	A(NODSF-5)/A(NODSF+5) - AVRGA)	NPARAYT	2926
TCOF(I,10)	= TREPS*((1. + GAMC(NFWD)*AM2(NFWD))*	NPARAYT	2927
Z	A(NODSF)/A(NODSF+5) - AVRGA)	NPARAYT	2928
IF (WCROS(2,ISLICE,NFWD+1).NE.0.0)	FILM =	NPARAYT	2929
Z	(WFC(NFWD+1)+WFC(NFWD+2))/WCROS(2,ISLICE,NFWD+1)	NPARAYT	2930
TCOF(I,11)	= TREPS*(-1. - GAMC(NFWD+1)*AM2(NFWD+1)*	NPARAYT	2931
Z	(1. + 2.*FF(NFWD+1)*DLX(NODSF+5)/DH(NFWD+1)+FILM) - AVRGA)	NPARAYT	2932
C		NPARAYT	2933
PUMP(NFWD+1)	= (3.14159265*WS/30.):**2*	NPARAYT	2934
Z	RR(NFWD+1)*(RR(NFWD+1)-RR(NFWD))	NPARAYT	2935
ROOT	= SQRT(32.2*GAMC(NFWD+1)*R*	NPARAYT	2936
Z	T(2,ISLICE,NODSF+5)*AM2(NFWD+1))	NPARAYT	2937
PUMTRM	= 0.0	NPARAYT	2938
IF (ROOT.NE.0.0)	PUMTRM = (3.14159265*WS/30.):**2*RR(NFWD+1)*	NPARAYT	2939
Z	(RR(NFWD+1)-RR(NFWD))*WCROS(2,ISLICE,NFWD+1)	NPARAYT	2940

	Z	/ (A (NODSF+5) *ROOT*144.*32.2)	NPARAYT 2941
C			NPARAYT 2942
C	PUMP HAS UNITS OF (IN**2/SEC**2);	ROOT HAS UNITS OF (PT/SEC)	NPARAYT 2943
C			NPARAYT 2944
	DUMTER = 0.0		NPARAYT 2945
	IF (ADUMP.GT.0.) DUMTER = - WDUMP**2*R*		NPARAYT 2946
Z	(T(2,ISLICE,NODSF-5)+T(2,ISLICE,NODSF))/		NPARAYT 2947
Z	((P(1,ISLICE,NFWD-1) + P(1,ISLICE,NFWD))*ADUMP*A(NODSF+5)*32.2)		NPARAYT 2948
	TCOF(I,20) = - PUMTRM + DUMTER + TRTRM		NPARAYT 2949
Z	- (1.0-TREPS)*(P(1,ISLICE,NFWD-1)*TCOF(I,9)		NPARAYT 2950
Z	+ P(1,ISLICE,NFWD)*TCOF(I,10)+P(1,ISLICE,NFWD+1)*TCOF(I,11))		NPARAYT 2951
900	CONTINUE		NPARAYT 2952
C			NPARAYT 2953
	RETURN		NPARAYT 2954
	END		NPARAYT 2955

C----	SOURCE.NPLENMP	NPLENMP	2956
	SUBROUTINE PLNUM(WXX,PXX,PTEXT,TXX,TTEXT)	NPLENMP	2957
C		NPLENMP	2958
C-	SOURCE.NPLENMP----	NPLENMP	2959
C	A SUBROUTINE TO COMPUTE PRESSURE DROP IN THE CENTRAL COOLANT PLENUM	NPLENMP	2960
C		NPLENMP	2961
C		NPLENMP	2962
C		NPLENMP	2963
C		NPLENMP	2964
C	-----	NPLENMP	2965
C	-	NPLENMP	2966
C	ARGUMENTS FOR THIS SUBROUTINE ARE:	NPLENMP	2967
C		NPLENMP	2968
C	WXX = MASS FLOW RATE INTO THIS SLICE, LBM/HR	NPLENMP	2969
C	PXX = AVERAGE STATIC PRESSURE FOR THIS PLENUM SLICE, PSIA,	NPLENMP	2970
C	CALCULATED IN THIS SUBROUTINE.	NPLENMP	2971
C	PTEXT = TOTAL PRESSURE IN AND TOTAL PRESSURE OUT, PSIA	NPLENMP	2972
C	TXX = AVERAGE STATIC TEMPERATURE FOR THIS PLENUM SLICE, (F),	NPLENMP	2973
C	CALCULATED IN THIS SUBROUTINE	NPLENMP	2974
C	TTEXT = TOTAL TEMP. IN AND TOTAL TEMP.OUT, (F)	NPLENMP	2975
C	-----	NPLENMP	2976
C		NPLENMP	2977
C		NPLENMP	2978
C	COMMON /RADL/ APLN(15), DPLN(15), RIN(15), ROUT(15),	NPLENMP	2979
C	Z PIN(15), TIN(15), W(15), WS	NPLENMP	2980
C		NPLENMP	2981
C	COMMON /TCO/ ADUMP, BTA, CD, CP,	NPLENMP	2982
C	Z GAM, PIN, R, SPAN, TOG,	NPLENMP	2983
C	Z WDUMP, WIN, AKC(15,80), AKW(15,80),	NPLENMP	2984
C	Z A(400), AJET(80), AM2(80), CNUM(80),	NPLENMP	2985
C	Z DH(80), DHP(80), DHJ(80),	NPLENMP	2986
C	Z DLX(400), PF(80), HC(80), HG(80),	NPLENMP	2987
C	Z P(2,15,80), PEXIT(15), PUMP(80), QG(80),	NPLENMP	2988
C	Z QSNK(80), RR(80), S(15), T(2,15,400),	NPLENMP	2989
C	Z TG(80), TAU(400), WFC(80),	NPLENMP	2990
C	Z WJ(15,80), WCROS(2,15,80), XN(80),	NPLENMP	2991
C	Z ICOR, IPILN, IHUB, ITIP,	NPLENMP	2992
C	Z ISBLOK, ISLICE, NBLKSZ, NSLICE,	NPLENMP	2993
C	Z NPWD, NSTA, IHC(80)	NPLENMP	2994
C		NPLENMP	2995
C		NPLENMP	2996
C	COMMON /TRANSNT/ RHOC, RHOM, SPHTC, SPHTM,	NPLENMP	2997
C	Z DLTIME, TYME, TEPS, TYHMAX	NPLENMP	2998
C		NPLENMP	2999
C	COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSFL(2),	NPLENMP	3000

Z	CTHPP(2), CTCON(2), CDEN(2), CSPHT(2), CGASC(2),	NPLENMP 3001
Z	CVISC(2), CRHOVG(2), IUNITS	NPLENMP 3002
C		NPLENMP 3003
	DIMENSION BETTA(20), B(20), AMC(20), SIGMA(20), TT1(20), F1(20),	NPLENMP 3004
1	Z1(15), CH(15)	NPLENMP 3005
	DIMENSION SV(3), XK(4), XL(4)	NPLENMP 3006
C		NPLENMP 3007
C	----FUNP(PF) IS THE EQUATION FOR DELTA P OVER LENGTH DX	NPLENMP 3008
C		NPLENMP 3009
	FUNP(PF) = (D1*RRP*(PP/R/TP-V1)/144.0-2.*PF*G2*TP/PP/AA/AA/DD)	NPLENMP 3010
Z	/ (1.-TP*G2/(PP*AA)**2+G1*CP	NPLENMP 3011
Z	*778.161*R*TP*TP*V1/(PP*AA)**2/PP)*DX	NPLENMP 3012
C		NPLENMP 3013
C	----FUNT(XK) IS THE EQUATION FOR DELTA T OVER LENGTH DX	NPLENMP 3014
C		NPLENMP 3015
	FUNT(XK) = (D2*RRP/CP+G1*R*(TP/PP/AA)**2*(XK/PP/DX))	NPLENMP 3016
Z	/ (1.+G1*R*TP/(PP*AA)**2)*DX	NPLENMP 3017
C		NPLENMP 3018
C	INITIALIZE	NPLENMP 3019
1	CONTINUE	NPLENMP 3020
	DIPTOL=0.005	NPLENMP 3021
	ACH=1.	NPLENMP 3022
	KSIG=0	NPLENMP 3023
	NCC=1	NPLENMP 3024
	IS=0	NPLENMP 3025
	KTR1=0	NPLENMP 3026
	W(ISLICE) = WXX	NPLENMP 3027
C		NPLENMP 3028
C	SAVE INLET TOTAL PRESSURE (PSIA) IN PIN AND INLET TOTAL	NPLENMP 3029
C	TEMPERATURE (F) IN TIN	NPLENMP 3030
C		NPLENMP 3031
	PIN(ISLICE) = PTEXT	NPLENMP 3032
	TIN(ISLICE) = TTEXT	NPLENMP 3033
	CH(ISLICE)=0.0	NPLENMP 3034
	ZED=Z1(ISLICE)*1.01	NPLENMP 3035
	IF (ZED.EQ.0.) ZED=.001	NPLENMP 3036
	IF (TIN(ISLICE).GT.-430.0) GO TO 5	NPLENMP 3037
3	TIN(ISLICE)=50.0	NPLENMP 3038
5	SIGB=0.0	NPLENMP 3039
C		NPLENMP 3040
C		NPLENMP 3041
7	NSTNS= 4	NPLENMP 3042
	SEGMTS=NSTNS-1	NPLENMP 3043
C		NPLENMP 3044
C		NPLENMP 3045
	T1=TIN(ISLICE)+460.0	NPLENMP 3046
	B(1)=T1	NPLENMP 3047
	BETA1=PIN(ISLICE)**2	NPLENMP 3048
	BETTA(1)=BETA1	NPLENMP 3049
	DX=S(ISLICE)/SEGMTS	NPLENMP 3050
	DXTMP=DX	NPLENMP 3051
	XXN=NSTNS	NPLENMP 3052
C		NPLENMP 3053
C		NPLENMP 3054
13	DR= (ROUT(ISLICE)-RIN(ISLICE))/SEGMTS	NPLENMP 3055
C		NPLENMP 3056
	CCOMPUTE CONSTANT TERMS-C1-C8 -	NPLENMP 3057
17	CONTINUE	NPLENMP 3058
	TTX=B(1)	NPLENMP 3059
		NPLENMP 3060

CALL GASTBL(TTY,C,CP,GAM,PD,R,XMU)	NPLENMP 3061
J=1	NPLENMP 3062
C6=.5*(GAM-1.0)	NPLENMP 3063
C1=GAM/C6	NPLENMP 3064
C	NPLENMP 3065
C	NPLENMP 3066
IF (WS) 21,19,21	NPLENMP 3067
C NO PUMPING	NPLENMP 3068
19 C3=0.0	NPLENMP 3069
GO TO 23	NPLENMP 3070
C	NPLENMP 3071
C PUMPING	NPLENMP 3072
C	NPLENMP 3073
21 C3=2.36695E-6*(WS**2)/(C1*R)	NPLENMP 3074
23 C8=32.17*GAM*R	NPLENMP 3075
C5=1.0/SQRT(C8)	NPLENMP 3076
C7=1.0/(32.17*C1*R)	NPLENMP 3077
IF (J.GT.1) GO TO 33	NPLENMP 3078
25 CONTINUE	NPLENMP 3079
C	NPLENMP 3080
C COMPUTE CHANNEL REYNOLDS NO. IF J = 1	NPLENMP 3081
C	NPLENMP 3082
REY = 12.0*W(ISLICE)/XMU*DPLN(ISLICE)/APLN(ISLICE)	NPLENMP 3083
C	NPLENMP 3084
C COMPUTE FRICTION FACTOR	NPLENMP 3085
C	NPLENMP 3086
C COMPUTE Z TERMS	NPLENMP 3087
33 CONTINUE	NPLENMP 3088
Z3=12.0*W(ISLICE)/XMU	NPLENMP 3089
Z4=(R*W(ISLICE)/3600.0)**2	NPLENMP 3090
IF (J.GT.1) GO TO 77	NPLENMP 3091
C DETERMINE INLET CONDITIONS	NPLENMP 3092
35 CONTINUE	NPLENMP 3093
C	NPLENMP 3094
C INITIAL STATION COMPUTATIONS -	NPLENMP 3095
C	NPLENMP 3096
C BALANCING ON TOTAL PRESSURE -	NPLENMP 3097
C	NPLENMP 3098
39 MAG=-1	NPLENMP 3099
41 SIGC=(B(J)/APLN(ISLICE))**2*Z4/BETTA(J)	NPLENMP 3100
IF (ABS(SIGB-SIGC).LE..00001*SIGC) GO TO 57	NPLENMP 3101
C SIGMA NOT CONVERGED	NPLENMP 3102
43 IS=IS+1	NPLENMP 3103
SV(IS)=SIGC	NPLENMP 3104
IF (IS.EQ.3) GO TO 135	NPLENMP 3105
45 B(J)=T1-C7*SIGC	NPLENMP 3106
IF (B(J).LT.50.0) GO TO 159	NPLENMP 3107
C TEMP OK	NPLENMP 3108
47 SIGB=SIGC	NPLENMP 3109
BETTA(J)=BETA1*(B(J)/T1)**C1	NPLENMP 3110
GO TO 41	NPLENMP 3111
C	NPLENMP 3112
C SIGMA CONVERGED	NPLENMP 3113
57 B(J)=T1-C7*SIGC	NPLENMP 3114
AMC(1)=SQRT(SIGC/B(J))*C5	NPLENMP 3115
IF (B(J).LE.0.0) GO TO 159	NPLENMP 3116
KTPBZ=0	NPLENMP 3117
63 BETTA(J)=BETA1/(1.0+C6*AMC(1)**2)**C1	NPLENMP 3118
IF (BETTA(J).LE.0.) GO TO 159	NPLENMP 3119
65 B(1)=T1/(1.0+C6*AMC(1)**2)	NPLENMP 3120

	SIGMA (1) = (B (J) /APLN (ISLICE)) **2 *Z4 /BETTA (J)	NPLENMP 3121
	SIGC=SQRT (SIGMA (1) /B (1)) *C5	NPLENMP 3122
	IF (ABS (SIGC-AMC (1)) .LE. .01) GO TO 71	NPLENMP 3123
67	AMC (1) =SIGC	NPLENMP 3124
69	KTRBZ=KTRBZ+1	NPLENMP 3125
	IF (KTRBZ .LE. 20) GO TO 63	NPLENMP 3126
71	TT1 (1) =TIN (ISLICE)	NPLENMP 3127
		NPLENMP 3128
		NPLENMP 3129
C	CHANNEL PRESSURE DROP -	NPLENMP 3130
		NPLENMP 3131
73	NAG = 1	NPLENMP 3132
	F1 (1) =REY	NPLENMP 3133
	SIGMA (2) =SIGMA (1) *.95	NPLENMP 3134
	J=2	NPLENMP 3135
	IS=0	NPLENMP 3136
77	CONTINUE	NPLENMP 3137
	AJ=J-1	NPLENMP 3138
	DR2=DR* (2.0* (RIN (ISLICE) +AJ*DR) -DR)	NPLENMP 3139
79	TT1 (J) =TT1 (J-1) +C3*DR2	NPLENMP 3140
	KSIG=0	NPLENMP 3141
	AZ=DPLN (ISLICE) /APLN (ISLICE)	NPLENMP 3142
	REY=Z3*AZ	NPLENMP 3143
	PP = SQRT (BETTA (1))	NPLENMP 3144
	SIGMA (1) =SQRT (SIGMA (1))	NPLENMP 3145
	BETTA (1) = PP	NPLENMP 3146
	TP = B (1)	NPLENMP 3147
	G2= (W (ISLICE) /3600.0) **2 *R /32.174	NPLENMP 3148
	G1= G2 /CP /778.161	NPLENMP 3149
	D1= (WS*3.1415927 /30.0) **2 *DR /DX /32.174	NPLENMP 3150
	D2= D1 /778.161 /144.0	NPLENMP 3151
	RRP = RIN (ISLICE)	NPLENMP 3152
	DO 97 J=2, NSTNS	NPLENMP 3153
	AZ=DPLN (ISLICE) /APLN (ISLICE)	NPLENMP 3154
	REY= Z3*AZ	NPLENMP 3155
	F1 (J) = .079*REY** (-.25)	NPLENMP 3156
	IF (REY.LT.2300.) F1 (J) = 16.0 /REY	NPLENMP 3157
	PTEMP=BETTA (J-1)	NPLENMP 3158
	TTEMP=B (J-1)	NPLENMP 3159
	RTEMP = RRP	NPLENMP 3160
	MACH1=1	NPLENMP 3161
	XNN=2.0	NPLENMP 3162
	DD=DPLN (ISLICE)	NPLENMP 3163
	AA=APLN (ISLICE)	NPLENMP 3164
	GO TO 85	NPLENMP 3165
81	MACH1=XNN	NPLENMP 3166
	WRITE (6,83) ISLICE,J	NPLENMP 3167
83	FORMAT (5X,9H***** ,50HDECREASED INCREMENT DERIVATIVE CHANGING	NPLENMP 3168
	1 TOO FAST,3X,'BRANCH NO. ',I2,', STATION NO. ',I2/)	NPLENMP 3169
	XNN=XNN*2.0	NPLENMP 3170
	DX=DX/2.0	NPLENMP 3171
	PTEMP=BETTA (J-1)	NPLENMP 3172
	TTEMP=B (J-1)	NPLENMP 3173
	PP = PTEMP	NPLENMP 3174
	TP = TTEMP	NPLENMP 3175
	RRP =RTEMP	NPLENMP 3176
	DD= DPLN (ISLICE)	NPLENMP 3177
	AA= APLN (ISLICE)	NPLENMP 3178
85	DO 91 L=1,4	NPLENMP 3179
	V1=G1 /PP /AA**2 / (1.0+G1 *R*TP / (PP*AA) **2)	NPLENMP 3180
	TERM1=TP*G2 / (PP*AA) **2	

	TERM2=G1*CP*778.161*R*TP*TP*V1/(PP*AA)**2/PP	NPLENMP 3181
	TESTMA=1.0-TERM1+TERM2	NPLENMP 3182
	IF (TESTMA.LE.0.0) GO TO 159	NPLENMP 3183
	XK(L)=FUNK(F1(J))	NPLENMP 3184
	IF (L.EQ.1) GO TO 89	NPLENMP 3185
	DO 87 LL=2,L	NPLENMP 3186
	XTEST=ABS((XK(L)-XK(LL-1))/PP)	NPLENMP 3187
	IF (XTEST.GT.DIFTOL) GO TO 81	NPLENMP 3188
87	CONTINUE	NPLENMP 3189
89	XL(L)=FUNK(XK(L))	NPLENMP 3190
	IF (L.EQ.4) GO TO 93	NPLENMP 3191
	PP=PTEMP+XK(L)/2.0	NPLENMP 3192
	TP=TTEMP+XL(L)/2.0	NPLENMP 3193
	IF (L.EQ.2) GO TO 91	NPLENMP 3194
	RRP=RRP+DR/XNN	NPLENMP 3195
	IF (L.NE.3) GO TO 91	NPLENMP 3196
	PP=PTEMP+XK(L)	NPLENMP 3197
	TP=TTEMP+XL(L)	NPLENMP 3198
91	CONTINUE	NPLENMP 3199
93	PP=PTEMP+XK(L)	NPLENMP 3200
	TP=TTEMP+XL(L)	NPLENMP 3201
	IF (PP.LE.0.0.OR.TP.LE.0.0) GO TO 159	NPLENMP 3202
	V1=G1/PP/AA**2/(1.0+G1*R*TP/(PP*AA)**2)	NPLENMP 3203
	TERM1=TP*G2/(PP*AA)**2	NPLENMP 3204
	TERM2=G1*CP*778.161*R*TP*TP*V1/(PP*AA)**2/PP	NPLENMP 3205
	TESTMA=1.0-TERM1+TERM2	NPLENMP 3206
	IF (TESTMA.LE.0.0) GO TO 159	NPLENMP 3207
	BETTA(J)=PTEMP+(XK(1)+2.0*(XK(2)+XK(3))+XK(4))/6.0	NPLENMP 3208
	B(J)=TTEMP+(XL(1)+2.0*(XL(2)+XL(3))+XL(4))/6.0	NPLENMP 3209
	PP=BETTA(J)	NPLENMP 3210
	IF (B(J).LE.0.0.OR.BETTA(J).LE.0.0) GO TO 159	NPLENMP 3211
	TP=B(J)	NPLENMP 3212
	IF (MACH1.EQ.1) GO TO 95	NPLENMP 3213
	MACH1=MACH1-1	NPLENMP 3214
	PTEMP=PP	NPLENMP 3215
	TTEMP=TP	NPLENMP 3216
	GO TO 85	NPLENMP 3217
95	XNN=2.0	NPLENMP 3218
	DX=DXTEMP	NPLENMP 3219
	SIGMA(J)=B(J)/APLN(ISLICE)/BETTA(J)*SQRT(Z4)	NPLENMP 3220
	AMC(J)=SIGMA(J)*SQRT(1.0/B(J))*C5	NPLENMP 3221
	IF (AMC(J).GE.1.0) GO TO 159	NPLENMP 3222
	F1(J)=REY	NPLENMP 3223
	TT1(J)=B(J)*(1.0+C6*AMC(J)**2)-460.0	NPLENMP 3224
97	CONTINUE	NPLENMP 3225
C	ALL STATIONS COMPUTED	NPLENMP 3226
99	SIGC=1.0	NPLENMP 3227
	AMC(NSTNS)=AMC(NSTNS)/SIGC	NPLENMP 3228
	IF (AMC(NSTNS).GT.1.0) GO TO 159	NPLENMP 3229
	BETTA(NSTNS)=BETTA(NSTNS)*SIGC**2	NPLENMP 3230
C	RESTART CHOKED BRANCH IF M.LT. .8	NPLENMP 3231
	IF (CH(ISLICE).EQ.0..OR.CH(ISLICE).EQ.1.) GO TO 113	NPLENMP 3232
	IF (ACH.EQ.(-1.0)) GO TO 113	NPLENMP 3233
	ACH=-1.	NPLENMP 3234
	AB=0.	NPLENMP 3235
	DO 109 J=1,NSTNS	NPLENMP 3236
	IF (AMC(J).GT..8) GO TO 113	NPLENMP 3237
	IF (AMC(J).LT.AB) GO TO 109	NPLENMP 3238
	AB=AMC(J)	NPLENMP 3239
109	CONTINUE	NPLENMP 3240

	CH(ISLICE)=0.0	NPLENMP 3241
	AJ=(GAM+1.)/2.	NPLENMP 3242
	CX=-(GAM+1.00)/(GAM-1.0)/2.0	NPLENMP 3243
	AZ=.95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX)	NPLENMP 3244
	WCHOKE=W(ISLICE)	NPLENMP 3245
	W(ISLICE)=AZ*W(ISLICE)	NPLENMP 3246
	WRITE(6,111) ISLICE,W(ISLICE),WCHOKE	NPLENMP 3247
111	FORMAT(8X,6H*****,23H RESTAPT CHOKED BRANCH ,I5,24H FLOW RATE INC	NPLENMP 3248
	1REASED TO ,F10.4,6H FROM ,F10.4,7H *****)	NPLENMP 3249
	GO TO 177	NPLENMP 3250
113	BETA1=PIN(ISLICE)**2	NPLENMP 3251
C	CALCULATE THE CHOKING FLOW RATE	NPLENMP 3252
	AB=0.0	NPLENMP 3253
	DO 115 J=1,NSTNS	NPLENMP 3254
	IF (AMC(J).LT.AB) GO TO 115	NPLENMP 3255
	AB=AMC(J)	NPLENMP 3256
115	CONTINUE	NPLENMP 3257
	AJ=(GAM+1.0)/2.0	NPLENMP 3258
	CX=-(GAM+1.0)/(GAM-1.0)/2.0	NPLENMP 3259
	AZ=.95/AB*(1.0+C6*.90)**(CX)*(1.0+C6*AB**2)**(-CX)	NPLENMP 3260
C	COMPUTE RESISTANCE EQUATION FOR BALANCE	NPLENMP 3261
	PT1=(1.0+C6*AMC(1)**2)**(C1/2.0)*BETTA(1)	NPLENMP 3262
	IF (C3.NE.0.0) GO TO 117	NPLENMP 3263
	GO TO 119	NPLENMP 3264
117	DR2=ROUT(ISLICE)**2-RIN(ISLICE)**2	NPLENMP 3265
119	PEXTT=PT1*(1.0+C3/T1*DR2)**(C1/2.0)	NPLENMP 3266
121	CONTINUE	NPLENMP 3267
	Z1(ISLICE)=(PEXTT**2-BETTA(NSTNS)**2)/W(ISLICE)**2	NPLENMP 3268
	IF (Z1(ISLICE).GT.0.0) GO TO 129	NPLENMP 3269
	WRITE(6,125) ISLICE,Z1(ISLICE)	NPLENMP 3270
125	FORMAT(///5X,'PASSAGE ',I3,5X,'HAS NEGATIVE OR NO RESISTANCE'	NPLENMP 3271
	Z ,F12.4//)	NPLENMP 3272
	Z1(ISLICE)=ZED	NPLENMP 3273
129	CONTINUE	NPLENMP 3274
	PP = BETTA(NSTNS)*(1.+C6*AMC(NSTNS)**2)**(C1/2.0)	NPLENMP 3275
	PTEXTIT = PP	NPLENMP 3276
	DIFTOL=0.005	NPLENMP 3277
	KTR1=0	NPLENMP 3278
C		NPLENMP 3279
C	COMPUTE AVERAGE STATIC PRESSURE AND STATIC TEMPERATURE	NPLENMP 3280
C		NPLENMP 3281
	PXX = 0.0	NPLENMP 3282
	TXX = 0.0	NPLENMP 3283
	DO 134 I = 1,NSTNS	NPLENMP 3284
	TXX = TXX + B(I)	NPLENMP 3285
134	PXX = PXX + BETTA(I)	NPLENMP 3286
	TXX = TXX/XXN - 460.	NPLENMP 3287
	PXX = PXX/XXN	NPLENMP 3288
	TTEXTIT = TT1(NSTNS)	NPLENMP 3289
	RETURN	NPLENMP 3290
C		NPLENMP 3291
C	CCOMPUTE ACCELERATION	NPLENMP 3292
C		NPLENMP 3293
135	D=SV(2)-SV(1)	NPLENMP 3294
137	D=(SV(3)-SV(2))/D	NPLENMP 3295
	E=ABS(D)-1.0	NPLENMP 3296
	IF (ABS(D).GT..6) GO TO 143	NPLENMP 3297
139	E=D/(D-1.0)	NPLENMP 3298
141	SIGB=E*SV(2)+(1.0-E)*SV(3)	NPLENMP 3299
143	IS=0	NPLENMP 3300

145	SIGC=SIGB	NPLENMP 3301
	IF (SIGC.LE.0.) GO TO 159	NPLENMP 3302
147	KSIG=KSIG+1	NPLENMP 3303
	IF (KSIG.LT.50) GO TO 155	NPLENMP 3304
149	WRITE (6,151) ISLICE,J	NPLENMP 3305
151	FORMAT(7X,28HPROGRAM IS LOOPING IN BRANCH,I6,9H STATION,I4//)	NPLENMP 3306
	KSIG=0	NPLENMP 3307
	NCC=NCC+1	NPLENMP 3308
	IF (NCC.LT.4) GO TO 159	NPLENMP 3309
153	ISLICE=220	NPLENMP 3310
	GO TO 129	NPLENMP 3311
155	CONTINUE	NPLENMP 3312
	IF (NAG.LT.0) GO TO 45	NPLENMP 3313
157	STOP	NPLENMP 3314
C		NPLENMP 3315
	CCHOKING ADJUSTMENT	NPLENMP 3316
C		NPLENMP 3317
159	WCHOKE=W(ISLICE)	NPLENMP 3318
	DIFTOL=0.050	NPLENMP 3319
161	IF (CH(ISLICE).EQ.0.) GO TO 165	NPLENMP 3320
163	W(ISLICE)=.98*W(ISLICE)	NPLENMP 3321
	IF (KTR1.GT.20) W(ISLICE)=W(ISLICE)*.98** (KTR1-20)	NPLENMP 3322
	GO TO 171	NPLENMP 3323
165	AB=0.	NPLENMP 3324
	IF (J.EQ.1) GO TO 169	NPLENMP 3325
	DO 167 IJK=1,J	NPLENMP 3326
	IF (AMC(IJK).GT.1.0) AMC(IJK)=1.0	NPLENMP 3327
	IF (AMC(IJK).LT.AB) GO TO 167	NPLENMP 3328
	AB=AMC(IJK)	NPLENMP 3329
167	CONTINUE	NPLENMP 3330
	AJ=(GAM+1.)/2.	NPLENMP 3331
	CX=-(GAM+1.00)/(GAM-1.0)/2.0	NPLENMP 3332
	AZ=.95/AB*(1.0+C6*.90)** (CX)*(1.0+C6*AB**2)** (-CX)	NPLENMP 3333
	W(ISLICE)=AZ*W(ISLICE)	NPLENMP 3334
	GO TO 171	NPLENMP 3335
169	CONTINUE	NPLENMP 3336
	W(ISLICE)=1600.0*APLN(ISLICE)*SQRT((32.17*BETA1*GAM)/(R*T1))	NPLENMP 3337
171	CH(ISLICE)=1.0	NPLENMP 3338
173	WRITE (6,175) ISLICE,J,W(ISLICE),WCHOKE	NPLENMP 3339
175	FORMAT(3X,12H*** PASSAGE ,I5,23H HAS CHOKED AT STATION ,I5,	NPLENMP 3340
	134H AND THE FLOW HAS BEEN REDUCED TO ,F10.4,6H FROM ,F10.4,4H ***)	NPLENMP 3341
177	IS=0	NPLENMP 3342
	BETTA (1)=PIN(ISLICE)**2	NPLENMP 3343
	B(1)=T1	NPLENMP 3344
	DX=DXTEMP	NPLENMP 3345
	J=1	NPLENMP 3346
	SIGB=0.0	NPLENMP 3347
	KTE1=KTR1+1	NPLENMP 3348
	IF (KTR1.GE.50) GO TO 181	NPLENMP 3349
179	GO TO 17	NPLENMP 3350
181	WRITE (6,183) ISLICE	NPLENMP 3351
183	FORMAT(/2X,16H***FLOW IN BRANCH,I6,	NPLENMP 3352
	Z ' HAS BEEN REDUCED 50 TIMES BECAUSE OF CHOKING')	NPLENMP 3353
	GO TO 153	NPLENMP 3354
	END	NPLENMP 3355
C		
	C-----SOURCE.NPLOTIT	NPLOTIT 3356
	SUBROUTINE PLOTMF (ALPH2)	NPLOTIT 3357
C		NPLOTIT 3358
C	SOURCE.NPLOTIT	NPLOTIT 3359
C		NPLOTIT 3360

```

COMMON /PRPS/ CPO,      GAMO,      DP(80),      SP(80),      RE(80),      NPLOTIT 3361
Z              CPC(80),   GAMC(80),   DUMR1(80),   DUMR2(80)      NPLOTIT 3362
C
COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000), NPLOTIT 3363
Z              IPLOT, MD1, MD2, MD3, IADJIN, IWRITE NPLOTIT 3364
C
COMMON /TCO/ ADUMF,      ETA,      CD,      CP,      NPLOTIT 3365
Z              GAM,      PIM,      E,      SPAN,      TOG, NPLOTIT 3366
Z              WDUMF,      WIM,      AKC(15,80), AKW(15,80), NPLOTIT 3367
Z              A(400), AJET(80), AM2(80), CNUM(80), NPLOTIT 3368
Z              DH(80), DHF(80), DHJ(80), NPLOTIT 3369
Z              DLX(400), PF(80), HC(80), HG(80), NPLOTIT 3370
Z              P(2,15,80), PEXIT(15), PUMP(80), QG(80), NPLOTIT 3371
Z              QSNK(80), RR(80), S(15), T(2,15,400), NPLOTIT 3372
Z              IG(80), TAU(400), WFC(80), NPLOTIT 3373
Z              WJ(15,80), WCROS(2,15,80), XN(80), NPLOTIT 3374
Z              ICOR, IFILM, IHUB, ITIF, NPLOTIT 3375
Z              ISBLOK, ISLICE, NBLKSZ, NSLICE, NPLOTIT 3376
Z              NFWF, NSTA, IHC(80) NPLOTIT 3377
C
COMMON /IRNSNT/ RHOC,      RHOM,      SPHTC,      SPHTM,      NPLOTIT 3378
Z              DLTME,      TYME,      TEPS,      TYMMAX NPLOTIT 3379
C
COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSPL(2), NPLOTIT 3380
Z              CTMPF(2), CTCON(2), CDEN(2), CSPHT(2), CGASC(2), NPLOTIT 3381
Z              CVISC(2), CRHOVG(2), IUNITS NPLOTIT 3382
C
DIMENSION Y(320), XLABL(29), YLABL(7), TLABL1(21), TLABL2(9) NPLOTIT 3383
DIMENSION XLABL2(15), VARIB(15), YPLABL(10), ALPH2(4), ALABL(7) NPLOTIT 3384
DIMENSION XS(80), XP(80), TSO(500), TSM(500), TPO(500), TPM(500) NPLOTIT 3385
DIMENSION YLAEL2(11) NPLOTIT 3386
DIMENSION PLEGN(5), SLEGN(5), SYMBL(10), YLBL(20), XLBL(20) NPLOTIT 3387
DIMENSION YTEM(80), ISYM(5), PLTYME(2) NPLOTIT 3388
LOGICAL*1 IXAX/.TRUE./, IYAX/.FALSE./ NPLOTIT 3389
INTEGER*2 NPTS NPLOTIT 3390
DIMENSION RTNARR(2), VARS(12), IVARS(12) NPLOTIT 3391
DATA PLEGN/'A) ', 'PRES', 'SURE', 'SID', 'E' '/' NPLOTIT 3392
DATA SLEGN/'B) ', 'SUCT', 'ION ', 'SID', 'E' '/' NPLOTIT 3393
DATA XLABL2/15* ' ' '/' NPLOTIT 3394
DATA SYMBL/'C'/' NPLOTIT 3395
DATA XLABL/'MID-', 'WALL', 'X/L', 'PRO', 'M ST', 'ATIO', 'N NO', '1, ' NPLOTIT 3396
Z              'L =', 'INC', 'HES' NPLOTIT 3397
Z              'SID', 'E', 'C', 'HNL', 'PL', 'ENUM', 'PRE' NPLOTIT 3398
Z              'SS.', 'PSI', 'A****' NPLOTIT 3399
DATA YLABL/'-----', 'TEMP', 'ERAT', 'URE', 'DEG', 'F.', '-----'/' NPLOTIT 3400
DATA YPLABL/' CO', 'OLAN', 'T CH', 'ANNE', 'L T', 'OTAL', 'PRE' NPLOTIT 3401
Z              'SSUR', 'E', 'PSIA'/' NPLOTIT 3402
DATA YLABL2/7* ' ', 'TEMP', 'ERAT', 'URE', 'F' '/' NPLOTIT 3403
DATA TLABL1/21* ' ' '/' NPLOTIT 3404
DATA TLABL2/9* ' ' '/' NPLOTIT 3405
DATA VARIB/'PRES', 'SURE', 'SUC', 'TION', 'SURF', 'ACE ', 'MID-', NPLOTIT 3406
Z              'WALL', 'CM ', 'KPA', '*****', 'K ', 'KP', 'A' '/' NPLOTIT 3407
DATA ALABL/2* ' ', '2*' NPLOTIT 3408
DATA SYMBL/'1', '2', '3', '4', '5', '6', '7', '8', '9', '0'/' NPLOTIT 3409
DATA ISYM/62,119,118,70,65/ NPLOTIT 3410
DATA PLTYME/2* ' '/' NPLOTIT 3411
ATYME = TYME NPLOTIT 3412
IF (ATYME.LT.0.0) ATYME = 0.0 NPLOTIT 3413
CALL NUMBER(4, ATYME, 8, 2, PLTYME) NPLOTIT 3414
ALABL(6) = PLTYME(1) NPLOTIT 3415
NPLOTIT 3416
NPLOTIT 3417
NPLOTIT 3418
NPLOTIT 3419
NPLOTIT 3420

```

```

      ALABL(7) = PLTYME(2)
C
C MD2 > 0 INDICATES JOB IS COMPLETE.--NOW DO SUMMARY PLOTS.
C
10  CONTINUE
    IF (MD2.GT.0) GO TO 80
20  NSTAPS = NSTA/2 + 1
    NLBLS = NSTAPS/5
C
C SET UP TIME AND DATE LABEL FOR PLOT IDENTIFICATION
C
      ALABL(1) = ALPH2(3)
      ALABL(2) = ALPH2(4)
      ALABL(3) = ALPH2(1)
      ALABL(4) = ALPH2(2)
C
C SET UP TITLE
C
      DO 45 I = 1,30
      IF (I.LE.21) TLABL1(I) = TITLE(I)
      IF (I.GT.21) TLABL2(I-21) = TITLE(I)
45  CONTINUE
C
C PRESSURE SIDE
C
46  IF (MD3.GT.1) GO TO 55
47  XP(1) = 0.0
    IX = 1
    DO 50 I = 3,NSTA,2
C
C NMM IS THE MIDMETAL NODE NUMBER (L)
C
      NMM = 5*I - 2
      IX = IX + 1
50  XP(IX) = XP(IX-1) + DLX(NMM)/CINCH(IUNITS)
      XPL = XP(NSTAPS)
      DO 51 I = 2,NSTAPS
51  XP(I) = XP(I)/XPL
55  CONTINUE
      IY = 0
      ITP = NSTAPS*(ISLICE-1)
      DO 60 I = 1,NSTA,2
      IY = IY + 1
      NOS = 5*I - 4
      Y(IY) = T(2,ISLICE,NOS)/CTMPF(IUNITS)
      IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.
      ITP = ITP + 1
      TPO(ITP) = Y(IY)
      IYP = IY + NSTAPS
      Y(IYP) = T(2,ISLICE,NOS+1)/CTMPF(IUNITS)
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
      IYP = IY + 2*NSTAPS
      Y(IYP) = T(2,ISLICE,NOS+2)/CTMPF(IUNITS)
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
      TPM(ITP) = Y(IYP)
      IYP = IY + 3*NSTAPS
      Y(IYP) = T(2,ISLICE,NOS+3)/CTMPF(IUNITS)
      IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
      IYP = IY + 4*NSTAPS
      MCOOL = NOS + 4

```

```

NPLOTIT 3421
NPLOTIT 3422
NPLOTIT 3423
NPLOTIT 3424
NPLOTIT 3425
NPLOTIT 3426
NPLOTIT 3427
NPLOTIT 3428
NPLOTIT 3429
NPLOTIT 3430
NPLOTIT 3431
NPLOTIT 3432
NPLOTIT 3433
NPLOTIT 3434
NPLOTIT 3435
NPLOTIT 3436
NPLOTIT 3437
NPLOTIT 3438
NPLOTIT 3439
NPLOTIT 3440
NPLOTIT 3441
NPLOTIT 3442
NPLOTIT 3443
NPLOTIT 3444
NPLOTIT 3445
NPLOTIT 3446
NPLOTIT 3447
NPLOTIT 3448
NPLOTIT 3449
NPLOTIT 3450
NPLOTIT 3451
NPLOTIT 3452
NPLOTIT 3453
NPLOTIT 3454
NPLOTIT 3455
NPLOTIT 3456
NPLOTIT 3457
NPLOTIT 3458
NPLOTIT 3459
NPLOTIT 3460
NPLOTIT 3461
NPLOTIT 3462
NPLOTIT 3463
NPLOTIT 3464
NPLOTIT 3465
NPLOTIT 3466
NPLOTIT 3467
NPLOTIT 3468
NPLOTIT 3469
NPLOTIT 3470
NPLOTIT 3471
NPLOTIT 3472
NPLOTIT 3473
NPLOTIT 3474
NPLOTIT 3475
NPLOTIT 3476
NPLOTIT 3477
NPLOTIT 3478
NPLOTIT 3479
NPLOTIT 3480

```


60	Y(IYP) = T(2,ISLICE,NCOOL)/CTMPF(IUNITS)	NPLOTIT 3481
	IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.	NPLOTIT 3482
	XLABL(16) = VARIB(1)	NPLOTIT 3483
	XLABL(17) = VARIB(2)	NPLOTIT 3484
	IF (IUNITS.EQ.2) GO TO 601	NPLOTIT 3485
	XLABL(12) = VARIB(9)	NPLOTIT 3486
	XLABL(13) = VARIB(10)	NPLOTIT 3487
	XLABL(28) = VARIB(11)	NPLOTIT 3488
	XLABL(29) = VARIB(12)	NPLOTIT 3489
	YLABL(5) = VARIB(13)	NPLOTIT 3490
	YLABL(6) = VARIB(10)	NPLOTIT 3491
	YPLABL(10) = VARIB(11)	NPLOTIT 3492
	YLABL2(11) = VARIB(13)	NPLOTIT 3493
601	CONTINUE	NPLOTIT 3494
	DO 611 I = 1, 15	NPLOTIT 3495
611	XLABL2(I) = XLABL(I+15)	NPLOTIT 3496
C		NPLOTIT 3497
	IF (IPLLOT.EQ.3) GO TO 63	NPLOTIT 3498
C		NPLOTIT 3499
	CPIM = PIM/CPRSR(IUNITS)	NPLOTIT 3500
	CALL NUMBER(1,ISLICE,4,0,XLABL2(6))	NPLOTIT 3501
	CALL NUMBER(4,CPIM,8,1,XLABL2(11))	NPLOTIT 3502
	CALL NUMBER(4,XPL,8,4,XLABL(10))	NPLOTIT 3503
	CALL CHARS(84,TLABL1,0.0,0.15,9.85,12)	NPLOTIT 3504
	CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)	NPLOTIT 3505
	CALL CHARS(60,XLABL,0.0,1.5,.25,12)	NPLOTIT 3506
	CALL CHARS(56,XLABL2,0.0,1.5,.05,12)	NPLOTIT 3507
	CALL CHARS(28,YLABL,90.,.25,3.3,12)	NPLOTIT 3508
	MD3 = MD3 + 1	NPLOTIT 3509
	CALL NUMBER(1,MD3,4,0,ALABL(5))	NPLOTIT 3510
	CALL CHARS(28,ALABL,0.0,6.2,9.3,12)	NPLOTIT 3511
C		NPLOTIT 3512
C---	TITLES ARE DONE, NOW SET UP AXES FOR TEMPERATURE PLOTS	NPLOTIT 3513
C		NPLOTIT 3514
	NPTS = NSTAPS	NPLOTIT 3515
	CALL SCALE(IXAX,NPTS,XP)	NPLOTIT 3516
	NPTS = 5*NSTAPS	NPLOTIT 3517
	CALL SCLBAK(IYAX,NPTS,Y,RTNAFR)	NPLOTIT 3518
	CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)	NPLOTIT 3519
	VAR(1) = 7.0	NPLOTIT 3520
	VAR(2) = 9.0	NPLOTIT 3521
	VAR(3) = 0.0	NPLOTIT 3522
	VAR(4) = 0.0	NPLOTIT 3523
	VAR(5) = 1.0	NPLOTIT 3524
	VAR(6) = .5	NPLOTIT 3525
	VAR(7) = 1.0	NPLOTIT 3526
	CALL XAXIS(.8,.6,VAR)	NPLOTIT 3527
	VAR(2) = 8.9	NPLOTIT 3528
	VAR(3) = 90.	NPLOTIT 3529
	VAR(4) = YMIN	NPLOTIT 3530
	VAR(5) = YMAX	NPLOTIT 3531
	CALL YAXIS(.8,.6,VAR)	NPLOTIT 3532
C		NPLOTIT 3533
C---	AXES ARE SET. NOW PLOT THE FIVE TEMPERATURE CURVES, USING	NPLOTIT 3534
C	DIFFERENT SYMBOLS FOR EACH.	NPLOTIT 3535
C		NPLOTIT 3536
	IVARS(1) = 4	NPLOTIT 3537
	IVARS(2) = NSTAPS	NPLOTIT 3538
	IVARS(3) = 2	NPLOTIT 3539
	DO 603 I = 1, 5	NPLOTIT 3540

IVARS(4) = ISYM(I)	NPLOTIT 3541
IYST = 1 + (I-1)*NSTAPS	NPLOTIT 3542
IYEN = I*NSTAPS	NPLOTIT 3543
III = 0	NPLOTIT 3544
DO 602 II = IYST,IYEN	NPLOTIT 3545
III = III + 1	NPLOTIT 3546
602 YTEM(III) = Y(II)	NPLOTIT 3547
CALL GPLOT(XP,YTEM,IVARS)	NPLOTIT 3548
603 CONTINUE	NPLOTIT 3549
CALL DISPLA(1)	NPLOTIT 3550
C	NPLOTIT 3551
C	NPLOTIT 3552
C PRESSURE SIDE COOLANT PRESSURE DISTRIBUTION	NPLOTIT 3553
C	NPLOTIT 3554
IY = 0	NPLOTIT 3555
DO 61 I = 1,NSTA,2	NPLOTIT 3556
IY = IY + 1	NPLOTIT 3557
61 Y(IY) = P(2,ISLICE,I)*	NPLOTIT 3558
Z (1.+(GAMC(I)-1.)*AM2(I)/2.)**(GAMC(I)/(GAMC(I)-1.))/CPRSR(IUNITS)	NPLOTIT 3559
CALL CHARS(84,TLABL1,0.0,0.15,9.85,12)	NPLOTIT 3560
CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)	NPLOTIT 3561
CALL CHARS(60,XLABL,0.0,1.5,.25,12)	NPLOTIT 3562
CALL CHARS(56,XLABL2,0.0,1.5,.05,12)	NPLOTIT 3563
CALL CHARS(40,YPLABL,90.,.25,2.8,12)	NPLOTIT 3564
MD3 = MD3 + 1	NPLOTIT 3565
CALL NUMBER(1,MD3,4,0,ALABL(5))	NPLOTIT 3566
CALL CHARS(28,ALABL,0.0,6.2,9.3,12)	NPLOTIT 3567
NPTS = NSTAPS	NPLOTIT 3568
CALL SCALE(IXAX,NPTS,XP)	NPLOTIT 3569
CALL SCLBAK(IYAX,NPTS,Y,RTNARR)	NPLOTIT 3570
CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)	NPLOTIT 3571
VAR(1) = 7.0	NPLOTIT 3572
VAR(2) = 9.0	NPLOTIT 3573
VAR(3) = 0.0	NPLOTIT 3574
VAR(4) = 0.0	NPLOTIT 3575
VAR(5) = 1.0	NPLOTIT 3576
VAR(6) = .5	NPLOTIT 3577
VAR(7) = 1.0	NPLOTIT 3578
CALL XAXIS(.8,.6,VAR)	NPLOTIT 3579
VAR(2) = 8.9	NPLOTIT 3580
VAR(3) = 90.	NPLOTIT 3581
VAR(4) = YMIN	NPLOTIT 3582
VAR(5) = YMAX	NPLOTIT 3583
CALL YAXIS(.8,.6,VAR)	NPLOTIT 3584
C	NPLOTIT 3585
C--- AXES ARE SET, NOW PLOT THE PRESSURE	NPLOTIT 3586
C	NPLOTIT 3587
IVARS(1) = 4	NPLOTIT 3588
IVARS(2) = NSTAPS	NPLOTIT 3589
IVARS(3) = 2	NPLOTIT 3590
IVARS(4) = 65	NPLOTIT 3591
CALL GPLOT(XP,Y,IVARS)	NPLOTIT 3592
CALL DISPLA(1)	NPLOTIT 3593
C	NPLOTIT 3594
C SUCTION SIDE	NPLOTIT 3595
C	NPLOTIT 3596
IF (MD3.GT.2) GO TO 69	NPLOTIT 3597
63 XS(1) = 0.0	NPLOTIT 3598
IX = 1	NPLOTIT 3599
DO 65 I = 2,NSTA,2	NPLOTIT 3600

```

NMM = 5*I - 2
IX = IX + 1
65 XS(IX) = XS(IX-1) + DLX(NMM)/CINCH(IUNITS)
XSL = XS(NSTAPS)
DO 66 I = 2,NSTAPS
66 XS(I) = XS(I)/XSL
69 CONTINUE
C

```

```

IY = 1
ITS = NSTAPS*(ISLICE-1) + 1
Y(IY) = T(2,ISLICE,1)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.
TSO(ITS) = Y(IY)
IYP = IY + NSTAPS
Y(IYP) = T(2,ISLICE,2)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
IYP = IYP + NSTAPS
Y(IYP) = T(2,ISLICE,3)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
TSM(ITS) = Y(IYP)
IYP = IYP + NSTAPS
Y(IYP) = T(2,ISLICE,4)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
IYP = IYP + NSTAPS
Y(IYP) = T(2,ISLICE,5)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
DO 70 I = 2,NSTA,2
IY = IY + 1
NOS = 5*I - 4
Y(IY) = T(2,ISLICE,NOS)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IY) = Y(IY) - 460.
ITS = ITS + 1
TSO(ITS) = Y(IY)
IYP = IY + NSTAPS
Y(IYP) = T(2,ISLICE,NOS+1)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
IYP = IYP + NSTAPS
Y(IYP) = T(2,ISLICE,NOS+2)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
TSM(ITS) = Y(IYP)
IYP = IYP + NSTAPS
Y(IYP) = T(2,ISLICE,NOS+3)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
IYP = IYP + NSTAPS
NCCOL = NOS + 4
70 Y(IYP) = T(2,ISLICE,NCCOL)/CTMPF(IUNITS)
IF (IUNITS.EQ.2) Y(IYP) = Y(IYP) - 460.
XLABL2(1) = VARIB(3)
XLABL2(2) = VARIB(4)
C

```

```

IF (IPLLOT.EQ.3) GO TO 80
C

```

```

CALL NUMBER(1,ISLICE,4,0,XLABL2(6))
CALL NUMBER(4,CPIM,8,1,XLABL2(11))
CALL NUMBER(4,XSL,8,4,XLABL(10))
IX = 2 + NFWD/2
CALL CHARS(84,TLABL1,0.0,0.15,9.85,12)
CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)
CALL CHARS(60,XLABL,0.0,1.5,.25,12)
CALL CHARS(56,XLABL2,0.0,1.5,.05,12)

```

```

NPLOTIT 3601
NPLOTIT 3602
NPLOTIT 3603
NPLOTIT 3604
NPLOTIT 3605
NPLOTIT 3606
NPLOTIT 3607
NPLOTIT 3608
NPLOTIT 3609
NPLOTIT 3610
NPLOTIT 3611
NPLOTIT 3612
NPLOTIT 3613
NPLOTIT 3614
NPLOTIT 3615
NPLOTIT 3616
NPLOTIT 3617
NPLOTIT 3618
NPLOTIT 3619
NPLOTIT 3620
NPLOTIT 3621
NPLOTIT 3622
NPLOTIT 3623
NPLOTIT 3624
NPLOTIT 3625
NPLOTIT 3626
NPLOTIT 3627
NPLOTIT 3628
NPLOTIT 3629
NPLOTIT 3630
NPLOTIT 3631
NPLOTIT 3632
NPLOTIT 3633
NPLOTIT 3634
NPLOTIT 3635
NPLOTIT 3636
NPLOTIT 3637
NPLOTIT 3638
NPLOTIT 3639
NPLOTIT 3640
NPLOTIT 3641
NPLOTIT 3642
NPLOTIT 3643
NPLOTIT 3644
NPLOTIT 3645
NPLOTIT 3646
NPLOTIT 3647
NPLOTIT 3648
NPLOTIT 3649
NPLOTIT 3650
NPLOTIT 3651
NPLOTIT 3652
NPLOTIT 3653
NPLOTIT 3654
NPLOTIT 3655
NPLOTIT 3656
NPLOTIT 3657
NPLOTIT 3658
NPLOTIT 3659
NPLOTIT 3660

```

CALL CHARS(28,YLABL,90.,.25,3.3,12)	NPLOTIT 3661
MD3 = MD3 + 1	NPLOTIT 3662
CALL NUMBER(1,MD3,4,0,ALABL(5))	NPLOTIT 3663
CALL CHARS(28,ALABL,0.0,6.2,9.3,12)	NPLOTIT 3664
C	NPLOTIT 3665
C--- TITLES ARE DONE, NOW SET UP AXES FOR TEMPERATURE PLOTS	NPLOTIT 3666
C	NPLOTIT 3667
NPTS = NSTAPS	NPLOTIT 3668
CALL SCALE(IXAX,NPTS,XS)	NPLOTIT 3669
NPTS = 5*NSTAPS	NPLOTIT 3670
CALL SCLBAK(IYAX,NPTS,Y,RTNARR)	NPLOTIT 3671
CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)	NPLOTIT 3672
VAR(1) = 7.0	NPLOTIT 3673
VAR(2) = 9.0	NPLOTIT 3674
VAR(3) = 0.0	NPLOTIT 3675
VAR(4) = 0.0	NPLOTIT 3676
VAR(5) = 1.0	NPLOTIT 3677
VAR(6) = .5	NPLOTIT 3678
VAR(7) = 1.0	NPLOTIT 3679
CALL XAXIS(.8,.6,VAR)	NPLOTIT 3680
VAR(2) = 8.9	NPLOTIT 3681
VAR(3) = 90.	NPLOTIT 3682
VAR(4) = YMIN	NPLOTIT 3683
VAR(5) = YMAX	NPLOTIT 3684
CALL YAXIS(.8,.6,VAR)	NPLOTIT 3685
C	NPLOTIT 3686
C--- AXES ARE SET. NOW PLOT THE FIVE TEMPERATURE CURVES, USING	NPLOTIT 3687
C DIFFERENT SYMBOLS FOR EACH.	NPLOTIT 3688
C	NPLOTIT 3689
IVAR(1) = 4	NPLOTIT 3690
IVAR(2) = NSTAPS	NPLOTIT 3691
IVAR(3) = 2	NPLOTIT 3692
DO 703 I = 1,5	NPLOTIT 3693
IVAR(4) = ISYM(I)	NPLOTIT 3694
IYST = 1 + (I-1)*NSTAPS	NPLOTIT 3695
IYEN = I*NSTAPS	NPLOTIT 3696
III = 0	NPLOTIT 3697
DO 702 II = IYST,IYEN	NPLOTIT 3698
III = III + 1	NPLOTIT 3699
702 YTEM(III) = Y(II)	NPLOTIT 3700
CALL GPLOT(XS,YTEM,IVAR)	NPLOTIT 3701
703 CONTINUE	NPLOTIT 3702
CALL DISPLA(1)	NPLOTIT 3703
C	NPLOTIT 3704
C	NPLOTIT 3705
C SUCTION SIDE COOLANT PRESSURE DISTRIBUTION	NPLOTIT 3706
C	NPLOTIT 3707
Y(1) = P(2,ISLICE,1)*	NPLOTIT 3708
Z (1.+(GAMC(1)-1.)*AM2(1)/2.)**(GAMC(1)/(GAMC(1)-1.))/CPRSR(IUNITS)	NPLOTIT 3709
IY = 1	NPLOTIT 3710
DO 75 I = 2,NSTA,2	NPLOTIT 3711
IY = IY + 1	NPLOTIT 3712
75 Y(IY) = P(2,ISLICE,I)*(1.+(GAMC(I)-1.)*	NPLOTIT 3713
Z *AM2(I)/2.)**(GAMC(I)/(GAMC(I)-1.))/CPRSR(IUNITS)	NPLOTIT 3714
CALL CHARS(84,TLABL1,0.0,0.15,9.85,12)	NPLOTIT 3715
CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)	NPLOTIT 3716
CALL CHARS(60,XLABL,0.0,1.5,.25,12)	NPLOTIT 3717
CALL CHARS(56,XLABL2,0.0,1.5,.05,12)	NPLOTIT 3718
CALL CHARS(40,YPLABL,90.,.25,2.8,12)	NPLOTIT 3719
MD3 = MD3 + 1	NPLOTIT 3720


```

CALL NUMBER(1,ND3,4,0,ALABL(5))
CALL CHARS(28,ALABL,0.0,6.2,9.3,12)
NPTS = NSTAPS
CALL SCALE(IXAX,NPTS,XS)
CALL SCLBAK(IYAX,NPTS,Y,RTNARR)
CALL GINTVL(RTNARR(1),RTNARR(2),10,1,YMIN,YMAX)
VARS(1) = 7.0
VARS(2) = 9.0
VARS(3) = 0.0
VARS(4) = 0.0
VARS(5) = 1.0
VARS(6) = .5
VARS(7) = 1.0
CALL XAXIS(.8,.6,VARS)
VARS(2) = 8.9
VARS(3) = 90.
VARS(4) = YMIN
VARS(5) = YMAX
CALL YAXIS(.8,.6,VARS)
C
C--- AXES ARE SET, NOW PLOT THE PRESSURE
C
  IVARS(1) = 4
  IVARS(2) = NSTAPS
  IVARS(3) = 2
  IVARS(4) = 65
  CALL GPLOT(XS,Y,IVARS)
  CALL DISPLA(1)
  GO TO 150
C
C THE FOLLOWING SECTION PUTS OUT PLOTS CONTAINING TEMPERATURES FROM
C ALL SLICES ON ONE FRAME
C
80  CONTINUE
    IF (ISLICE.LT.NSLICE) GO TO 150
C
C THE FOLLOWING PUTS TWO PLOTS ON ONE FRAME OF FILM
C
C FIRST PLOT THE OUTSIDE SURFACE TEMPERATURES FOR EACH
C SLICE ON THE SAME PLOT
C
  NPTS = NSTAPS*NSLICE
  CALL SCLBAK(IYAX,NPTS,TPC,RTNARR)
  THAXP = RTNARR(2)
  THINP = RTNARR(1)
  CALL SCLBAK(IYAX,NPTS,TSO,RTNARR)
  THAXS = RTNARR(2)
  THINS = RTNARR(1)
  IF (THAXS.GT.THAXP) THAXP = THAXS
  IF (THINS.LT.THINP) THINP = THINS
  NINTRV = (THAXP-THINP)/100. + 2
  CALL GINTVL(THINP,THAXP,NINTRV,0,ATMINP,ATHAXP)
  AINTRV = NINTRV
  CALL CHAPS(84,TLABL1,0.0,0.15,9.85,12)
  CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)
  YLABL2(5) = VARIB(5)
  YLABL2(6) = VARIB(6)
  CALL CHARS(44,YLABL2,90.,.25,1.6,12)
  VARS(1) = 8.
  VARS(2) = 8.5

```

```

NPLOTTIT 3721
NPLOTTIT 3722
NPLOTTIT 3723
NPLOTTIT 3724
NPLOTTIT 3725
NPLOTTIT 3726
NPLOTTIT 3727
NPLOTTIT 3728
NPLOTTIT 3729
NPLOTTIT 3730
NPLOTTIT 3731
NPLOTTIT 3732
NPLOTTIT 3733
NPLOTTIT 3734
NPLOTTIT 3735
NPLOTTIT 3736
NPLOTTIT 3737
NPLOTTIT 3738
NPLOTTIT 3739
NPLOTTIT 3740
NPLOTTIT 3741
NPLOTTIT 3742
NPLOTTIT 3743
NPLOTTIT 3744
NPLOTTIT 3745
NPLOTTIT 3746
NPLOTTIT 3747
NPLOTTIT 3748
NPLOTTIT 3749
NPLOTTIT 3750
NPLOTTIT 3751
NPLOTTIT 3752
NPLOTTIT 3753
NPLOTTIT 3754
NPLOTTIT 3755
NPLOTTIT 3756
NPLOTTIT 3757
NPLOTTIT 3758
NPLOTTIT 3759
NPLOTTIT 3760
NPLOTTIT 3761
NPLOTTIT 3762
NPLOTTIT 3763
NPLOTTIT 3764
NPLOTTIT 3765
NPLOTTIT 3766
NPLOTTIT 3767
NPLOTTIT 3768
NPLOTTIT 3769
NPLOTTIT 3770
NPLOTTIT 3771
NPLOTTIT 3772
NPLOTTIT 3773
NPLOTTIT 3774
NPLOTTIT 3775
NPLOTTIT 3776
NPLOTTIT 3777
NPLOTTIT 3778
NPLOTTIT 3779
NPLOTTIT 3780

```

VAR(3) = 0.	NPLOTIT 3781
VAR(4) = 0.0	NPLOTIT 3782
VAR(5) = 1.0	NPLOTIT 3783
VAR(6) = .5	NPLOTIT 3784
VAR(7) = 1.0	NPLOTIT 3785
VAR(8) = 0.0	NPLOTIT 3786
CALL XAXIS(1.2,5.5,VAR)	NPLOTIT 3787
MD3 = MD3 + 1	NPLOTIT 3788
CALL NUMBER(1,MD3,4,0,ALAB(5))	NPLOTIT 3789
CALL CHARS(28,ALAB,0.0,1.3,9.5,12)	NPLOTIT 3790
CALL CHARS(20,PLEGN,0.0,6.0,9.5,12)	NPLOTIT 3791
VAR(1) = 7.	NPLOTIT 3792
VAR(2) = 3.8	NPLOTIT 3793
VAR(3) = 90.	NPLOTIT 3794
VAR(4) = ATMINP	NPLOTIT 3795
VAR(5) = ATMAXP	NPLOTIT 3796
VAR(6) = AINTRV	NPLOTIT 3797
VAR(7) = 1.	NPLOTIT 3798
CALL YAXIS(1.2,5.5,VAR)	NPLOTIT 3799
DO 100 I = 1,NSLICE	NPLOTIT 3800
JST = NSTAPS*(I-1)	NPLOTIT 3801
DO 95 J = 1,NSTAPS	NPLOTIT 3802
95 Y(J) = TPO(JST+J)	NPLOTIT 3803
SYMBOL = SYMBL(I)	NPLOTIT 3804
KS = 0	NPLOTIT 3805
C	NPLOTIT 3806
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO	NPLOTIT 3807
C IDENTIFY THE CURVES.	NPLOTIT 3808
KSTART = I + 1	NPLOTIT 3809
DO 98 K = KSTART,NSTAPS,10	NPLOTIT 3810
KS = KS + 1	NPLOTIT 3811
XLBL(KS) = XP(K)	NPLOTIT 3812
98 YLBL(KS) = Y(K)	NPLOTIT 3813
IVARS(1) = 6	NPLOTIT 3814
IVARS(2) = NLBLS	NPLOTIT 3815
IVARS(3) = 3	NPLOTIT 3816
IVARS(4) = 240+I	NPLOTIT 3817
IVARS(5) = 1	NPLOTIT 3818
IVARS(6) = 8	NPLOTIT 3819
CALL GPLOT(XLBL,YLBL,IVARS)	NPLOTIT 3820
IVARS(1) = 3	NPLOTIT 3821
IVARS(2) = NSTAPS	NPLOTIT 3822
IVARS(3) = 0	NPLOTIT 3823
100 CALL GPLOT(XP,Y,IVARS)	NPLOTIT 3824
C	NPLOTIT 3825
C	NPLOTIT 3826
VAR(1) = 7.	NPLOTIT 3827
VAR(2) = 8.5	NPLOTIT 3828
VAR(3) = 0.	NPLOTIT 3829
VAR(4) = 0.0	NPLOTIT 3830
VAR(5) = 1.0	NPLOTIT 3831
VAR(6) = .5	NPLOTIT 3832
VAR(7) = 1.0	NPLOTIT 3833
CALL XAXIS(1.2,.5,VAR)	NPLOTIT 3834
CALL CHARS(42,XLBL,0.0,3.0,.05,12)	NPLOTIT 3835
VAR(1) = 7.	NPLOTIT 3836
VAR(2) = 3.8	NPLOTIT 3837
VAR(3) = 90.	NPLOTIT 3838
VAR(4) = ATMINP	NPLOTIT 3839
VAR(5) = ATMAXP	NPLOTIT 3840

VAR5(6) = AINTRV	NPLOTIT 3841
VAR5(7) = 1.	NPLOTIT 3842
CALL YAXIS(1.2,.5,VAR5)	NPLOTIT 3843
CALL CHARS(20,SLEGN,0.0,6.0,4.5,12)	NPLOTIT 3844
DO 110 I = 1,NSLICE	NPLOTIT 3845
JST = NSTAPS*(I-1)	NPLOTIT 3846
DO 105 J = 1,NSTAPS	NPLOTIT 3847
105 Y(J) = TSO(JST+J)	NPLOTIT 3848
SYMBOL = SYMBL(I)	NPLOTIT 3849
KS = 0	NPLOTIT 3850
C	NPLOTIT 3851
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO	NPLOTIT 3852
C IDENTIFY THE CURVES.	NPLOTIT 3853
KSTART = I + 1	NPLOTIT 3854
DO 108 K = KSTART,NSTAPS,10	NPLOTIT 3855
KS = KS + 1	NPLOTIT 3856
108 XLEL(KS) = XS(K)	NPLOTIT 3857
YLBL(KS) = Y(K)	NPLOTIT 3858
IVARS(1) = 6	NPLOTIT 3859
IVARS(2) = NLBLS	NPLOTIT 3860
IVARS(3) = 3	NPLOTIT 3861
IVARS(4) = 240+I	NPLOTIT 3862
IVARS(5) = 1	NPLOTIT 3863
IVARS(6) = 8	NPLOTIT 3864
CALL GPLOT(XLEL,YLBL,IVARS)	NPLOTIT 3865
IVARS(1) = 3	NPLOTIT 3866
IVARS(2) = NSTAPS	NPLOTIT 3867
IVARS(3) = 0	NPLOTIT 3868
110 CALL GPLOT(XS,Y,IVARS)	NPLOTIT 3869
CALL DISPLA(1)	NPLOTIT 3870
C	NPLOTIT 3871
C	NPLOTIT 3872
C	NPLOTIT 3873
C NOW PLOT THE MID-METAL TEMPERATURES FOR EACH SLICE, ALL ON ONE PLOT	NPLOTIT 3874
C	NPLOTIT 3875
112 CONTINUE	NPLOTIT 3876
CALL SCLBAK(IYAX,NPTS,TPM,RTNARR)	NPLOTIT 3877
TMAXP = RTNARR(2)	NPLOTIT 3878
TMINP = RTNARR(1)	NPLOTIT 3879
CALL SCLBAK(IYAX,NPTS,TSM,RTNARR)	NPLOTIT 3880
TMAXS = RTNARR(2)	NPLOTIT 3881
TMINS = RTNARR(1)	NPLOTIT 3882
IF (TMAXS.GT.TMAXP) TMAXP = TMAXS	NPLOTIT 3883
IF (TMINS.LT.TMINP) TMINP = TMINS	NPLOTIT 3884
NINTRV = (TMAXP-TMINP)/100. + 2	NPLOTIT 3885
CALL GINTVL(TMINP,TMAXP,NINTRV,0,ATMINP,ATMAXP)	NPLOTIT 3886
AINTRV = NINTRV	NPLOTIT 3887
CALL CHAFS(84,TLABL1,0.0,0.15,9.85,12)	NPLOTIT 3888
CALL CHARS(36,TLABL2,0.0,0.15,9.65,12)	NPLOTIT 3889
YLABL2(5) = VARIB(7)	NPLOTIT 3890
YLABL2(6) = VARIB(8)	NPLOTIT 3891
CALL CHARS(44,YLABL2,90.,.25,1.6,12)	NPLOTIT 3892
VAR5(1) = 8.	NPLOTIT 3893
VAR5(2) = 8.5	NPLOTIT 3894
VAR5(3) = 0.	NPLOTIT 3895
VAR5(4) = 0.0	NPLOTIT 3896
VAR5(5) = 1.0	NPLOTIT 3897
VAR5(6) = .5	NPLOTIT 3898
VAR5(7) = 1.0	NPLOTIT 3899
VAR5(8) = 0.0	NPLOTIT 3900

CALL XAXIS(1.2,5.5,VAR5)	NPLOTIT 3901
MD3 = MD3 + 1	NPLOTIT 3902
CALL NUMBER(1,MD3,4,0,ALABL(5))	NPLOTIT 3903
CALL CHARS(28,ALABL,0.0,1.3,9.5,12)	NPLOTIT 3904
CALL CHARS(20,PLEGN,0.0,6.0,9.5,12)	NPLOTIT 3905
VAR5(1) = 7.	NPLOTIT 3906
VAR5(2) = 3.8	NPLOTIT 3907
VAR5(3) = 90.	NPLOTIT 3908
VAR5(4) = ATMIMP	NPLOTIT 3909
VAR5(5) = ATMAXP	NPLOTIT 3910
VAR5(6) = AINTRV	NPLOTIT 3911
VAR5(7) = 1.	NPLOTIT 3912
CALL YAXIS(1.2,5.5,VAR5)	NPLOTIT 3913
114 CONTINUE	NPLOTIT 3914
DO 120 I = 1,NSLICE	NPLOTIT 3915
JST = NSTAPS*(I-1)	NPLOTIT 3916
DO 115 J = 1,NSTAPS	NPLOTIT 3917
115 Y(J) = TPM(JST+J)	NPLOTIT 3918
SYMBOL = SYMBL(I)	NPLOTIT 3919
KS = 0	NPLOTIT 3920
C	NPLOTIT 3921
C--- LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO	NPLOTIT 3922
C IDENTIFY THE CURVES.	NPLOTIT 3923
KSTART = I + 1	NPLOTIT 3924
DO 118 K = KSTART,NSTAPS,10	NPLOTIT 3925
KS = KS + 1	NPLOTIT 3926
XLBL(KS) = XP(K)	NPLOTIT 3927
118 YLBL(KS) = Y(K)	NPLOTIT 3928
IVARS(1) = 6	NPLOTIT 3929
IVARS(2) = NLBLS	NPLOTIT 3930
IVARS(3) = 3	NPLOTIT 3931
IVARS(4) = 240+I	NPLOTIT 3932
IVARS(5) = 1	NPLOTIT 3933
IVARS(6) = 8	NPLOTIT 3934
CALL GLOT(XLBL,YLBL,IVARS)	NPLOTIT 3935
IVARS(1) = 3	NPLOTIT 3936
IVARS(2) = NSTAPS	NPLOTIT 3937
IVARS(3) = 0	NPLOTIT 3938
120 CALL GLOT(XP,Y,IVARS)	NPLOTIT 3939
C	NPLOTIT 3940
C	NPLOTIT 3941
VAR5(1) = 7.	NPLOTIT 3942
VAR5(2) = 8.5	NPLOTIT 3943
VAR5(3) = 0.	NPLOTIT 3944
VAR5(4) = 0.0	NPLOTIT 3945
VAR5(5) = 1.0	NPLOTIT 3946
VAR5(6) = .5	NPLOTIT 3947
VAR5(7) = 1.0	NPLOTIT 3948
CALL XAXIS(1.2,.5,VAR5)	NPLOTIT 3949
CALL CHARS(42,XLABL,0.0,3.5,.05,15)	NPLOTIT 3950
VAR5(1) = 7.	NPLOTIT 3951
VAR5(2) = 3.8	NPLOTIT 3952
VAR5(3) = 90.	NPLOTIT 3953
VAR5(4) = ATMIMP	NPLOTIT 3954
VAR5(5) = ATMAXP	NPLOTIT 3955
VAR5(6) = AINTRV	NPLOTIT 3956
VAR5(7) = 1.	NPLOTIT 3957
CALL YAXIS(1.2,.5,VAR5)	NPLOTIT 3958
CALL CHARS(20,SLEGN,0.0,6.0,4.5,12)	NPLOTIT 3959
DO 130 I = 1,NSLICE	NPLOTIT 3960

	JST = NSTAPS*(I-1)	NPLOTIT 3961
	DO 125 J = 1,NSTAPS	NPLOTIT 3962
125	Y(J) = TSM(JST+J)	NPLOTIT 3963
	SYMBOL = SYMBL(I)	NPLOTIT 3964
	KS = 0	NPLOTIT 3965
C		NPLOTIT 3966
C---	LABEL EVERY 10TH POINT WITH THE SLICE NUMBER, TO	NPLOTIT 3967
C	IDENTIFY THE CURVES.	NPLOTIT 3968
	KSTART = I + 1	NPLOTIT 3969
	DO 128 K = KSTART,NSTAPS,10	NPLOTIT 3970
	KS = KS + 1	NPLOTIT 3971
	XLBL(KS) = XS(K)	NPLOTIT 3972
128	YLBL(KS) = Y(K)	NPLOTIT 3973
	IVARS(1) = 6	NPLOTIT 3974
	IVARS(2) = NLBLS	NPLOTIT 3975
	IVARS(3) = 3	NPLOTIT 3976
	IVARS(4) = 240+I	NPLOTIT 3977
	IVARS(5) = 1	NPLOTIT 3978
	IVARS(6) = 8	NPLOTIT 3979
	CALL GPLOTT(XLBL,YLBL,IVARS)	NPLOTIT 3980
	IVARS(1) = 3	NPLOTIT 3981
	IVARS(2) = NSTAPS	NPLOTIT 3982
	IVARS(3) = 0	NPLOTIT 3983
130	CALL GPLOTT(XS,Y,IVARS)	NPLOTIT 3984
	CALL DISPLA(1)	NPLOTIT 3985
C		NPLOTIT 3986
C		NPLOTIT 3987
150	CONTINUE	NPLOTIT 3988
	RETURN	NPLOTIT 3989
	END	NPLOTIT 3990
C----	SOURCE.NPREPAT	NPREPAT 3991
	SUBROUTINE PREP(ICHNL,NTTG)	NPREPAT 3992
C		NPREPAT 3993
C-	SOURCE.NPREPAT----	NPREPAT 3994
C		NPREPAT 3995
	COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGP(1000),	NPREPAT 3996
Z	BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),	NPREPAT 3997
Z	BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),	NPREPAT 3998
Z	QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80),	NPREPAT 3999
Z	QTIPIN(400), QTIP(80), RHOVG(400), PEX(400),	NPREPAT 4000
Z	BCTIME(50), TTIO(50), PTIO(50), WPLEN,	NPREPAT 4001
Z	WSVST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCP	NPREPAT 4002
C		NPREPAT 4003
	COMMON /FLMCOL/ RHOVGA(80), PG(80), XFC(80), FLMEFF(80),	NPREPAT 4004
Z	XMUC(80), LMES(80), REFC(80), NFCSUP(80)	NPREPAT 4005
C		NPREPAT 4006
	COMMON /FRIC/ ALPHA, BETA, DELTA, EPS	NPREPAT 4007
C		NPREPAT 4008
	COMMON /PRPS/ CPO, GANO, DP(80), SP(80), RE(80),	NPREPAT 4009
Z	CPC(80), GAMC(80), DUMR1(80), DUMR2(80)	NPREPAT 4010
C		NPREPAT 4011
	COMMON /SPECL/ CHANL(8000), TITLE(30), INDCHN(2000),	NPREPAT 4012
Z	IPLLOT, MD1, MD2, MD3, IADJIN, IWRITE	NPREPAT 4013
C		NPREPAT 4014
	COMMON /TCC/ ADUMP, ETA, CD, CP,	NPREPAT 4015
Z	GAM, PIM, R, SPAN, TOG,	NPREPAT 4016
Z	WDUMP, WIM, AKC(15,80), AKW(15,80),	NPREPAT 4017
Z	A(400), AJET(80), AM2(80), CNUM(80),	NPREPAT 4018
Z	DH(80), DHF(80), DHJ(80),	NPREPAT 4019
Z	DLX(400), FP(80), HC(80), HG(80),	NPREPAT 4020

Z	P(2,15,80),	PEXIT(15),	PUMP(80),	QG(80),	NPREPAT 4021
Z	QSNK(80),	RR(80),	S(15),	T(2,15,400),	NPREPAT 4022
Z	TG(80),	TAU(400),	WFC(80),		NPREPAT 4023
Z	WJ(15,80),	WCROS(2,15,80),	XN(80),		NPREPAT 4024
Z	ICOR,	IFILM,	IHUB,	ITIP,	NPREPAT 4025
Z	ISBLOK,	ISLICE,	NBLKSZ,	NSLICE,	NPREPAT 4026
Z	NFWD,	NSTA,	IHC(80)		NPREPAT 4027
C					NPREPAT 4028
	COMMON /TENSNT/	RHOC,	PHOM,	SPHTC,	NPREPAT 4029
	Z	DLTYME,	TYME,	TEPS,	NPREPAT 4030
C				SPHIM,	NPREPAT 4031
C				TYMMAX	NPREPAT 4032
C					NPREPAT 4033
C	ICHNL IS THE CHANNEL NUMBER; = 1 FOR THE HUB REGION,				NPREPAT 4034
C		= NSLICE AT THE TIP			NPREPAT 4035
C					NPREPAT 4036
C					NPREPAT 4037
C	-- LOCATE INPUT DATA FOR THIS CHANNEL AND STORE IT IN WORKING ARRAYS.				NPREPAT 4038
C					NPREPAT 4039
C	I1 IS THE STARTING POINT IN THE INDCHN ARRAY FOR THIS CHANNEL				NPREPAT 4040
C					NPREPAT 4041
	I1 = INDCHN(ICHNL)				NPREPAT 4042
	IF (ICHNL.NE.INDCHN(I1)) GO TO 290				NPREPAT 4043
C					NPREPAT 4044
C	-- IF ABOVE TEST IS TRUE, THEN THE DATA IS NOT STORED WHERE EXPECTED				NPREPAT 4045
C					NPREPAT 4046
10	CONTINUE				NPREPAT 4047
	IFILM = INDCHN(I1+1)				NPREPAT 4048
	ICOR = INDCHN(I1+2)				NPREPAT 4049
	NFWD = INDCHN(I1+3)				NPREPAT 4050
	NSTA = INDCHN(I1+4)				NPREPAT 4051
	ISBLOK = INDCHN(I1+5)				NPREPAT 4052
	NBLKSZ = INDCHN(I1+6)				NPREPAT 4053
	IPLT = INDCHN(I1+7)				NPREPAT 4054
	MD1 = INDCHN(I1+8)				NPREPAT 4055
	MD2 = INDCHN(I1+9)				NPREPAT 4056
	IHUB = INDCHN(I1+12)				NPREPAT 4057
	ITIP = INDCHN(I1+13)				NPREPAT 4058
	IN1 = I1 + 14				NPREPAT 4059
	CD = CHANL(ISBLOK)				NPREPAT 4060
	ALPHA = CHANL(ISBLOK+1)				NPREPAT 4061
	BETA = CHANL(ISBLOK+2)				NPREPAT 4062
	DELTA = CHANL(ISBLOK+3)				NPREPAT 4063
	EPS = CHANL(ISBLOK+4)				NPREPAT 4064
	ADUMP = CHANL(ISBLOK+6)				NPREPAT 4065
	SPAN = CHANL(ISBLOK+7)				NPREPAT 4066
	S(ICHNL) = SPAN				NPREPAT 4067
	BTA = CHANL(ISBLOK+8)				NPREPAT 4068
	DLTYME = CHANL(ISBLOK+9)				NPREPAT 4069
	TEPS = CHANL(ISBLOK+10)				NPREPAT 4070
	NODSF = 5*NFWD				NPREPAT 4071
	NODST = 5*NSTA				NPREPAT 4072
	I1 = ISBLOK + 14				NPREPAT 4073
	I3 = ISBLOK + 14 + 2*NODST				NPREPAT 4074
C*****					NPREPAT 4075
12	CONTINUE				NPREPAT 4076
	DO 205 I = 1,NODST				NPREPAT 4077
	IM = I1 + I				NPREPAT 4078
	TAU(I) = CHANL(IM)				NPREPAT 4079
	A(I) = TAU(I)*SPAN				NPREPAT 4080

205	IM = IM + NODST	NPREPAT 4081
	DLX(I) = CHANL(IM)	NPREPAT 4082
	DO 215 I = 1,NSTA	NPREPAT 4083
	IFLU = 5*I	NPREPAT 4084
	DH(I) = 4.0*A(IFLU)/(2.*(SPAN+TAU(IFLU)))	NPREPAT 4085
	IM = I3 + I	NPREPAT 4086
	DHJ(I) = CHANL(IM)	NPREPAT 4087
	IM = IM + NSTA	NPREPAT 4088
	DHF(I) = CHANL(IM)	NPREPAT 4089
	IM = IM + NSTA	NPREPAT 4090
	XN(I) = CHANL(IM)	NPREPAT 4091
14	CONTINUE	NPREPAT 4092
	IF (DHJ(I).GT.0.0.AND.XN(I).GT.0.0) GO TO 202	NPREPAT 4093
	AJET(I) = 0.0	NPREPAT 4094
	GO TO 212	NPREPAT 4095
202	CONTINUE	NPREPAT 4096
	XOD=XN(I)/DHJ(I)	NPREPAT 4097
	IF(XOD.LT.3.1.OR.XOD.GT.12.5) WRITE(6,527) I,XOD	NPREPAT 4098
527	FORMAT(1H0,' WARNING, RATIO OF JET HOLE SPACING TO JET DIAMETER	NPREPAT 4099
	1'FOR JET ',I2,' IS',F10.4,' WHICH IS OUT OF'	NPREPAT 4100
	2' THE RANGE OF VALIDITY OF THE CORRELATION.')	NPREPAT 4101
	CNUM(I)=SPAN/XN(I)	NPREPAT 4102
C	CNUM IS THE NUMBER OF IMPINGEMENT JETS AT CHANNEL NODE I	NPREPAT 4103
C	TOTAL JET AREA IS (AREA OF ONE JET)*(NUMBER OF JETS)	NPREPAT 4104
C		NPREPAT 4105
54	AJET(I)=3.14159*DHJ(I)**2/4.*CNUM(I)	NPREPAT 4106
212	CONTINUE	NPREPAT 4107
	IM = IM + NSTA	NPREPAT 4108
	FR(I) = CHANL(IM)	NPREPAT 4109
	IM = IM + NSTA	NPREPAT 4110
	DP(I) = CHANL(IM)	NPREPAT 4111
	IM = IM + NSTA	NPREPAT 4112
	SP(I) = CHANL(IM)	NPREPAT 4113
	IM = IM + NSTA	NPREPAT 4114
	IM = IM + NSTA	NPREPAT 4115
	INN = IN1 + I	NPREPAT 4116
	IHC(I) = INDCHN(INN)	NPREPAT 4117
215	CONTINUE	NPREPAT 4118
C*****		NPREPAT 4119
C		NPREPAT 4120
C--	NOW, GIVEN SLICE, ICHNL, EVALUATE B.C. AT METAL NODE POINTS. IN THE	NPREPAT 4121
C	FOLLOWING:	NPREPAT 4122
C--	XS & XP ARE DISTANCE FROM LEADING EDGE, ALONG OUTSIDE SURFACE	NPREPAT 4123
C	(INCHES), FOR SUCTION & PRESSURE SIDES.	NPREPAT 4124
C--	THE CONVENTIONS USED IN THE FOLLOWING ARE: INDEX BEGINING WITH -I-	NPREPAT 4125
C--	IS A SLICE INDEX, INDEX BEGINING WITH -N- IS A TIME INDEX, INDEX	NPREPAT 4126
C--	BEGINING WITH -L- IS A N X INDEX, AND AN INDEX BEGINING WITH -J- IS	NPREPAT 4127
C--	A PROPERTY INDEX I.E. HG,QG,TG,PG.	NPREPAT 4128
C		NPREPAT 4129
C--	FIRST, CHECK THAT THIS IS A TRANSIENT CASE, AND DETERMINE THE MAX.	NPREPAT 4130
C--	BCTIME INDEX, NMX.	NPREPAT 4131
	NMX = 1	NPREPAT 4132
310	IF (BCTIME(NMX+1).LE.0.0) GO TO 315	NPREPAT 4133
	NMX = NMX + 1	NPREPAT 4134
	GO TO 310	NPREPAT 4135
315	CONTINUE	NPREPAT 4136
C		NPREPAT 4137
C--	NOW, IF THIS IS A TRANSIENT, FIND THE LOCATION IN THE BCTIME ARRAY	NPREPAT 4138
C--	OF THE CURRENT TIME, AND CALCULATE THE VALUE OF THE INTERPOLATING	NPREPAT 4139
C--	PARAMETER, TMFRAC.	NPREPAT 4140

TMFRAC = 0.0	NPREPAT 4141
NLST = 1	NPREPAT 4142
IF (NMX.EQ.1.OR.TYME.LE.0.0) GO TO 330	NPREPAT 4143
C-- THE ABOVE TRANSFER OCCURS IF THIS IS A STEADY STATE PROBLEM	NPREPAT 4144
C-- THE FOLLOWING TRANSFER OCCURS IF WE ARE BEYOND THE LAST BCTIME ENTRY	NPREPAT 4145
NLST = NMX	NPREPAT 4146
IF (TYME.GE.BCTIME(NMX)) GO TO 330	NPREPAT 4147
NMXM1 = NMX - 1	NPREPAT 4148
DO 320 N = 1, NMXM1	NPREPAT 4149
NLST = N	NPREPAT 4150
IF (TYME.GE.BCTIME(N).AND.TYME.LT.BCTIME(N+1)) GO TO 325	NPREPAT 4151
320 CONTINUE	NPREPAT 4152
325 TMFRAC = (TYME-BCTIME(NLST))/(BCTIME(NLST+1)-BCTIME(NLST))	NPREPAT 4153
C	NPREPAT 4154
C-- NEXT, SEARCH THE BCXS & BCXP ARRAYS TO FIND THE X INTERPOLATING	NPREPAT 4155
C-- FACTORS, XSF & XPF, FOR POSITIONS XS & XP, SLICE ICHNL.	NPREPAT 4156
C-- THE BRACKETING INDICES ARE LBLWS & LABOVS AND LBLWP & LABOVP.	NPREPAT 4157
C	NPREPAT 4158
C-- THE STARTING POINTS IN THE BCXS & BCXP ARRAYS FOR THIS SLICE ARE:	NPREPAT 4159
C	NPREPAT 4160
330 LSS = (ICHNL-1)*NBCS	NPREPAT 4161
LSP = (ICHNL-1)*NBCP	NPREPAT 4162
XS = 0.0	NPREPAT 4163
XP = 0.0	NPREPAT 4164
C	NPREPAT 4165
C-- THE STARTING POINTS IN THE PROPERTY ARRAYS, FOR THE LATEST TIME STEP	NPREPAT 4166
C-- ARE GIVEN BY:	NPREPAT 4167
C	NPREPAT 4168
JS1S = NSLICE*(NLST-1)*NBCS + LSS	NPREPAT 4169
JS1P = NSLICE*(NLST-1)*NBCP + LSP	NPREPAT 4170
IF (NMX.EQ.NLST) GO TO 335	NPREPAT 4171
JS2S = JS1S + NSLICE*NBCS	NPREPAT 4172
JS2P = JS1P + NSLICE*NBCP	NPREPAT 4173
335 HG(1) = BCHGS(JS1S+1) + TMFRAC*(BCHGS(JS2S+1)-BCHGS(JS1S+1))	NPREPAT 4174
TG(1) = BCTGS(JS1S+1) + TMFRAC*(BCTGS(JS2S+1)-BCTGS(JS1S+1))	NPREPAT 4175
QG(1) = BCQGS(JS1S+1) + TMFRAC*(BCQGS(JS2S+1)-BCQGS(JS1S+1))	NPREPAT 4176
PG(1) = BCPGS(JS1S+1) + TMFRAC*(BCPGS(JS2S+1)-BCPGS(JS1S+1))	NPREPAT 4177
C	NPREPAT 4178
DO 350 K = 2, NSTA, 2	NPREPAT 4179
C-- THE OUTSIDE SURFACE NODE NUMBERS FOR S & P SIDES ARE:	NPREPAT 4180
NNS = 5*K - 4	NPREPAT 4181
NNP = 5*K + 1	NPREPAT 4182
XS = XS + DLX(NNS)	NPREPAT 4183
XP = XP + DLX(NNP)	NPREPAT 4184
C	NPREPAT 4185
DO 340 L = 1, NBCS	NPREPAT 4186
LBLWS = LSS + L - 1	NPREPAT 4187
LABOVS = LSS + L	NPREPAT 4188
IF (BCXS(LABOVS).GT.XS) GO TO 342	NPREPAT 4189
340 CONTINUE	NPREPAT 4190
C	NPREPAT 4191
C-- INSERT ERROR MESSAGE HERE-- EXTRAPOLATING BEYOND THE BCXS TABLE	NPREPAT 4192
C	NPREPAT 4193
342 XSF = (XS-BCXS(LBLWS))/(BCXS(LABOVS)-BCXS(LBLWS))	NPREPAT 4194
C	NPREPAT 4195
DO 345 L = 1, NBCP	NPREPAT 4196
LBLWP = LSP + L - 1	NPREPAT 4197
LABOVP = LSP + L	NPREPAT 4198
IF (BCXP(LABOVP).GT.XP) GO TO 347	NPREPAT 4199
345 CONTINUE	NPREPAT 4200

C		NPREPAT 4201
C--	INSERT ERROR MESSAGE HERE-- EXTRAPOLATING BEYOND THE BCXP TABLE	NPREPAT 4202
C		NPREPAT 4203
347	XPF = (XP-BCXP(LBLOWP))/(BCXP(LABOV)-BCXP(LBLOWP))	NPREPAT 4204
C		NPREPAT 4205
C--	NOW THE FRACTIONS ARE KNOWN, CALCULATE THE INTERPOLATED PROPERTIES.	NPREPAT 4206
C--	FIRST, FOR THE STEADY STATE OR FOR TIMES BEYOND THE LAST BCTIME:	NPREPAT 4207
C		NPREPAT 4208
	JB1S = NSLICE*(NLST-1)*NBCS + LBLOWS	NPREPAT 4209
	JB1P = NSLICE*(NLST-1)*NBCP + LBLOWP	NPREPAT 4210
	HG(K) = BCHGS(JB1S) + XSF*(BCHGS(JB1S+1)-BCHGS(JB1S))	NPREPAT 4211
	HG(K+1) = BCHGP(JB1P) + XPF*(BCHGP(JB1P+1)-BCHGP(JB1P))	NPREPAT 4212
	QG(K) = BCQGS(JB1S) + XSF*(BCQGS(JB1S+1)-BCQGS(JB1S))	NPREPAT 4213
	QG(K+1) = BCQGP(JB1P) + XPF*(BCQGP(JB1P+1)-BCQGP(JB1P))	NPREPAT 4214
	TG(K) = BCTGS(JB1S) + XSF*(BCTGS(JB1S+1)-BCTGS(JB1S)) + 460.	NPREPAT 4215
	TG(K+1) = BCTGP(JB1P) + XPF*(BCTGP(JB1P+1)-BCTGP(JB1P)) + 460.	NPREPAT 4216
	PG(K) = BCPGS(JB1S) + XSF*(BCPGS(JB1S+1)-BCPGS(JB1S))	NPREPAT 4217
	PG(K+1) = BCPGP(JB1P) + XPF*(BCPGP(JB1P+1)-BCPGP(JB1P))	NPREPAT 4218
C		NPREPAT 4219
	IF (NMX.EQ.1.OR.TYME.GE.BCTIME(NMX).OR.TYME.LE.0.0) GO TO 350	NPREPAT 4220
	JB2S = NSLICE*NLST*NBCS + LBLOWS	NPREPAT 4221
	JB2P = NSLICE*NLST*NBCP + LBLOWP	NPREPAT 4222
	AHG = BCHGS(JB2S) + XSF*(BCHGS(JB2S+1)-BCHGS(JB2S))	NPREPAT 4223
	HG(K) = HG(K) + TMFRAC*(AHG-HG(K))	NPREPAT 4224
	AHG = BCHGP(JB2P) + XPF*(BCHGP(JB2P+1)-BCHGP(JB2P))	NPREPAT 4225
	HG(K+1) = HG(K+1) + TMFRAC*(AHG-HG(K+1))	NPREPAT 4226
	AQG = BCQGS(JB2S) + XSF*(BCQGS(JB2S+1)-BCQGS(JB2S))	NPREPAT 4227
	QG(K) = QG(K) + TMFRAC*(AQG-QG(K))	NPREPAT 4228
	AQG = BCQGP(JB2P) + XPF*(BCQGP(JB2P+1)-BCQGP(JB2P))	NPREPAT 4229
	QG(K+1) = QG(K+1) + TMFRAC*(AQG-QG(K+1))	NPREPAT 4230
	ATG = BCTGS(JB2S) + XSF*(BCTGS(JB2S+1)-BCTGS(JB2S)) + 460.	NPREPAT 4231
	TG(K) = TG(K) + TMFRAC*(ATG-TG(K))	NPREPAT 4232
	ATG = BCTGP(JB2P) + XPF*(BCTGP(JB2P+1)-BCTGP(JB2P)) + 460.	NPREPAT 4233
	TG(K+1) = TG(K+1) + TMFRAC*(ATG-TG(K+1))	NPREPAT 4234
	APG = BCPGS(JB2S) + XSF*(BCPGS(JB2S+1)-BCPGS(JB2S))	NPREPAT 4235
	PG(K) = PG(K) + TMFRAC*(APG-PG(K))	NPREPAT 4236
	APG = BCPGP(JB2P) + XPF*(BCPGP(JB2P+1)-BCPGP(JB2P))	NPREPAT 4237
	PG(K+1) = PG(K+1) + TMFRAC*(APG-PG(K+1))	NPREPAT 4238
350	CONTINUE	NPREPAT 4239
	RETURN	NPREPAT 4240
290	WRITE(6,295) ICHNL	NPREPAT 4241
295	FORMAT(// ' CHANNEL NO. ',I3,' DATA STORAGE IS MESSED UP',	NPREPAT 4242
Z	' *%><::%%*%&@&\$?***%***')	NPREPAT 4243
149	CONTINUE	NPREPAT 4244
	RETURN	NPREPAT 4245
	END	NPREPAT 4246
C----	SOURCE.NTARAYT	NTARAYT 4247
	SUBROUTINE TARRAY(JS,JSSENS,DELTAN)	NTARAYT 4248
C		NTARAYT 4249
C-	SOURCE.NTARAYT----	NTARAYT 4250
C		NTARAYT 4251
C		NTARAYT 4252
C++++	A SUBROUTINE TO SET UP THE COEFFICIENT ARRAY TO SOLVE FOR	NTARAYT 4253
C	BLADE TEMPERATURES, TRANSIENT CALCULATIONS.	NTARAYT 4254
C	FIRST PUT TOGETHER FROM STEADY STATE PROGRAM, NOVEMBER 24, 1975	NTARAYT 4255
C		NTARAYT 4256
C		NTARAYT 4257
C*****	*****	NTARAYT 4258
C*****	*****	NTARAYT 4259
C		NTARAYT 4260

```
C
C
C   NODE NAMES FOR STATION IS ARE:
C     LCOOL = COOLANT NODE AT IS.
C     LIN   = INSIDE WALL NODE.
C     L     = MID WALL NODE.
C     LOUO  = OUTSIDE WALL NODE.
C     LCUP  = ADJACENT UPSTREAM COOLANT NODE.
C     LCUPS = ADJACENT INSIDE WALL NODE
C     LUP   = ADJACENT UPSTREAM MID-WALL NODE.
C     LDN   = ADJACENT DOWNSTREAM MID-WALL NODE.
C     LJ    = JUNCTION OF COATING AND WALL METAL
C
C                                     IS
C                                     |
C      -----LOUT-*-----
C      -----LJ *-----
C      //////////////////////////////////////
C      //LUP-*/L-*/LDN-*/
C      //////////////////////////////////////
C      -LCUPS-*---LIN-*---
C
C FLOW
C --->  LCUP-*    LCOOL-*
C
C      -LCUPP-*--LCOOLP-*-----
C
C      PRESSURE SIDE WALL OR PLENUM
C*****
C*****
C
C REAL*8 TCOF
C COMMON /BOUND/ BCXS(400), BCPX(400), BCHGS(1000), BCHGP(1000),
Z          BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),
Z          BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),
Z          QHUBIN(400), QHUB(80), TTIPIN(400), TTIP(80),
Z          QTIPIN(400), QTIP(80), RHOVG(400), PEX(400),
Z          BCTIME(50), TTIO(50), PTIO(50), WPLEN,
Z          WSVST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCEP
C
C COMMON /FLMCOL/ RHOVGA(80), PG(80), XFC(80), FLMEFF(80),
Z          XMUC(80), EMES(80), REFC(80), NFCSUP(80)
C
C COMMON /MATRX/ TCOF(400,30)
C
C COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),
Z          CPC(80), GAMC(80), DUMR1(80), DUMR2(80)
C
C COMMON /TCO/ ADUMP, BTA, CD, CP,
Z          GAM, PIM, E, SPAN, TOG,
Z          WDUMP, WIM, AKC(15,80), AKW(15,80),
Z          A(400), AJET(80), AM2(80), CNUM(80),
Z          DH(80), DHF(80), DHJ(80),
Z          DLX(400), FF(80), HC(80), HG(60),
Z          P(2,15,80), PEXIT(15), PUMP(80), QG(80),
Z          QSNK(80), RR(80), S(15), T(2,15,400),
Z          TG(80), TAU(400), WPC(80),
Z          WJ(15,80), WCROS(2,15,80),
Z          ICOR, IFILM, IHUB, ITIF,
```

Z	ISBLOK,	ISLICE,	NBLKSZ,	NSLICE,	NTARAYT 4321
Z	NFWD,	NSTA,	IHC(80)		NTARAYT 4322
C	COMMON /TRNSNT/	RHOC,	RHOM,	SPHTC,	NTARAYT 4323
Z	DLTYME,	TYME,	TEPS,	SPHTM,	NTARAYT 4324
				TYMMAX	NTARAYT 4325
	DIMENSION EFAREA(80), DELTAN(15)				NTARAYT 4326
C	RTRNVN = 0.				NTARAYT 4327
	RCHRDN = 0.				NTARAYT 4328
	TREPS = 1.0				NTARAYT 4329
	IF (TYME.GE.0.) TREPS = TEPS				NTARAYT 4330
	HX = 1.0				NTARAYT 4331
	RCVRY = .89				NTARAYT 4332
300	CONTINUE				NTARAYT 4333
	SPAN = S(ISLICE)				NTARAYT 4334
	NODST = 5*NSTA				NTARAYT 4335
	NODSF = 5*NFWD				NTARAYT 4336
	DO 308 J = 1,30				NTARAYT 4337
	DO 308 I = 1,400				NTARAYT 4338
308	TCOF(I,J) = 0.0				NTARAYT 4339
	DO 309 I = 1,9				NTARAYT 4340
	ICO = NODST + I				NTARAYT 4341
309	DLX(ICO) = 0.0				NTARAYT 4342
	ICOMS = JS + 2 - 4*JSSENS				NTARAYT 4343
	ICOMP = JS - 2 + 4*JSSENS				NTARAYT 4344
	C****ICOMS IS STATION ADJACENT TO JS, IN SUCTION DIRECTION				NTARAYT 4345
	C****ICOMP IS STATION ADJACENT TO JS, IN PRESSURE DIRECTION				NTARAYT 4346
	IF (ICOMS.LT.0) ICOMS=2				NTARAYT 4347
	IF (ICOMP.LT.1) ICOMP=1				NTARAYT 4348
310	CONTINUE				NTARAYT 4349
C					NTARAYT 4350
C**	BEGIN OVERALL LOOP, WHERE LOOP VARIABLE (IS) IS THE STATION NUMBER				NTARAYT 4351
C					NTARAYT 4352
	DO 440 IS = 1,NSTA				NTARAYT 4353
	ISUP = IS - 2				NTARAYT 4354
	YIMP = 0.0				NTARAYT 4355
	YFINS = 0.0				NTARAYT 4356
	YCONV = 0.0				NTARAYT 4357
	YIMPU = 0.0				NTARAYT 4358
	YFINSU = 0.0				NTARAYT 4359
	YCONVU = 0.0				NTARAYT 4360
	YIMPUU = 0.0				NTARAYT 4361
	YFNSUU = 0.0				NTARAYT 4362
	YCNVUU = 0.0				NTARAYT 4363
	IF (IHC(IS).EQ.1) YIMP=1.0				NTARAYT 4364
	IF (IHC(IS).EQ.2) YCONV= (1.0+RCVRY*AN2(IS)*(GAMC(IS)-1.)/2.)				NTARAYT 4365
	IF (IHC(IS).EQ.3) YFINS=1.0				NTARAYT 4366
	FACTOR = 1.0				NTARAYT 4367
	IF (IS.EQ.ICOMS.OR.IS.EQ.ICOMP) FACTOR = .5				NTARAYT 4368
	ISENS = 0				NTARAYT 4369
	ISEN = IS - 2*(IS/2)				NTARAYT 4370
C					NTARAYT 4371
C	IF (IS.GT.NFWD+1)-- IN TRAILING EDGE REGION, GO TO 380				NTARAYT 4372
C					NTARAYT 4373
	IF(IS.GT.NFWD+1) GO TO 380				NTARAYT 4374
C					NTARAYT 4375
	J9 = 16				NTARAYT 4376
	LCOOL = 5*IS				NTARAYT 4377
	LIN =LCOOL-1				NTARAYT 4378
	L =LCOOL-2				NTARAYT 4379
					NTARAYT 4380

LJ = LCOOL - 3	NTARAYT 4381
LOUT = LCOOL - 4	NTARAYT 4382
IF (IHC(IS).NE.2) GO TO 320	NTARAYT 4383
C	NTARAYT 4384
C FORCED CONVECTION HC:	NTARAYT 4385
THP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.	NTARAYT 4386
CALL GASTBL(THP,C,CP,GAM,PD,R,XMU)	NTARAYT 4387
RE(IS) = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(A(LCOOL)*XMU)	NTARAYT 4388
HC(IS) = .023*12.*(C/DH(IS))*(RE(IS)**.8)*(PD**.333)	NTARAYT 4389
C	NTARAYT 4390
320 CONTINUE	NTARAYT 4391
IF (IS.GE.JS) GO TO 322	NTARAYT 4392
C	NTARAYT 4393
C SPECIAL CASE FOR STATION NUMBER 1:	NTARAYT 4394
IF (IS.EQ.1) ISUP = 2	NTARAYT 4395
C	NTARAYT 4396
C IF STATION IS FORWARD OF FLOW SPLIT, AND ON SAME SIDE, GO TO 370	NTARAYT 4397
IF (ISEN.EQ.JSENS) GO TO 370	NTARAYT 4398
C	NTARAYT 4399
322 CONTINUE	NTARAYT 4400
IF (ISUP.LT.1) ISUP = 1	NTARAYT 4401
IF (IHC(ISUP).EQ.1) YIMPU = 1.0	NTARAYT 4402
IF (IHC(ISUP).EQ.2) YCONVU = 1.0 + RCVRY*AM2(ISUP)*(GAMC(ISUP)-1.)/2.	NTARAYT 4403
IF (IHC(ISUP).EQ.3) YFINSU = 1.0	NTARAYT 4404
LCUP = LCOOL - 10	NTARAYT 4405
LCUPS = LCOOL - 11	NTARAYT 4406
LUP = LCOOL - 12	NTARAYT 4407
LDN = LCOOL + 8	NTARAYT 4408
IF (IS.NE.2) GO TO 324	NTARAYT 4409
C	NTARAYT 4410
C IF THIS IS STATION NUMBER 2:	NTARAYT 4411
LCUP = 5	NTARAYT 4412
LCUPS = 4	NTARAYT 4413
LUP = 3	NTARAYT 4414
324 CONTINUE	NTARAYT 4415
IF (IS.GT.1) GO TO 326	NTARAYT 4416
C	NTARAYT 4417
C IF THIS IS STATION NUMBER 1:	NTARAYT 4418
LCUP = 10	NTARAYT 4419
LCUPS = 9	NTARAYT 4420
LUP = 8	NTARAYT 4421
326 CONTINUE	NTARAYT 4422
C	NTARAYT 4423
C IS = 1, STATION NO. 1, LEADING EDGE NODES	NTARAYT 4424
C	NTARAYT 4425
IF (IS.NE.JS) GO TO 330	NTARAYT 4426
C	NTARAYT 4427
C*****	NTARAYT 4428
C*****	NTARAYT 4429
C***** THIS BLOCK COMPUTES TCOF ELEMENTS FOR THE STATION AT WHICH	NTARAYT 4430
C THE FLOW SPLITS	NTARAYT 4431
C***** IS = JS	NTARAYT 4432
C*****	NTARAYT 4433
C*****	NTARAYT 4434
C	NTARAYT 4435
328 CONTINUE	NTARAYT 4436
DX1 = DLX(LUP)	NTARAYT 4437
IF (DX1.EQ.0.0) DX1 = DLX(L)	NTARAYT 4438
DX2 = DLX(LDN)	NTARAYT 4439
IF (DX2.EQ.0.0) DX2 = DLX(L)	NTARAYT 4440


```

DX3 = DLX(LUP-2)
IF (DX3.EQ.0.0) DX3 = DLX(LOUT)
DX4 = DLX(LDN-2)
IF (DX4.EQ.0.0) DX4 = DLX(LOUT)
DX5 = DLX(LUP+1)
IF (DX5.EQ.0.0) DX5 = DLX(LIN)
DX6 = DLX(LDN+1)
IF (DX6.EQ.0.0) DX6 = DLX(LIN)
DX9 = DLX(LUP-1)
IF (DX9.EQ.0.0) DX9 = DLX(LJ)
DX10 = DLX(LDN-1)
IF (DX10.EQ.0.0) DX10 = DLX(LJ)
CURV = 1.0 + (DX9+DX10)/(DX3+DX4)

C
C CURV IS A MEASURE OF THE CURVATURE OF THE BLADE AT STATION IS.
C CURV = 2.0 IS A STRAIGHT SECTION OF WALL,
C CURV < 2.0 IS A CONVEX SECTION, AND
C CURV > 2.0 IS A CONCAVE SECTION
C

TRTRMC = 0.0
IF (DLTYME.GT.0.0.AND.TYME.GE.0.0) TRTRMC=(3600./144.)*RHOC*SPHTC*
Z (TAU(LOUT)**2)/(4.*AKC(ISLICE,IS)*DLTYME)
TCOF(LOUT,13) = -TREPS + TRTRMC*CURV
TCOF(LOUT,12) = TRTRMC*CURV + TREPS*(1.0+(1.0-BTA)*HG(IS)*TAU(LOUT)/
Z (12.*AKC(ISLICE,IS)))
TCOF(LOUT,J9) = -(1.-BTA)*TREPS*FLMEFF(IS)*HG(IS)*TAU(LOUT)/
Z (12.*AKC(ISLICE,IS))
TCOF(LOUT,24) = (BTA*QG(IS) + (1.0-BTA)*HG(IS)*TG(IS)*
Z (1.0-FLMEFF(IS))*TAU(LOUT)/(12.*AKC(ISLICE,IS))
Z - T(1,ISLICE,LOUT)*((1.-TREPS)*((1.-BTA)*HG(IS)*TAU(LOUT)/
Z (12.*AKC(ISLICE,IS)) + 1.) - TRTRMC*CURV)
Z + T(1,ISLICE,LJ)*(1.-TREPS)*TRTRMC*CURV
Z + T(1,ISLICE,LCOOL)*(1.-TREPS)*FLMEFF(IS)*(1.-BTA)*HG(IS)*
Z TAU(LOUT)/(12.*AKC(ISLICE,IS))

C
TCOF(LJ,11) = TREPS
TCOF(LJ,13) = TREPS*(AKW(ISLICE,IS)/AKC(ISLICE,IS))*
Z (2.*TAU(LOUT)/TAU(L))*(DX1+DX2+DX3+DX4)/(DX9+DX10+DX3+DX4)
TCOF(LJ,12) = -TCOF(LJ,11) - TCOF(LJ,13)
TCOF(LJ,24) = (1.-TREPS)*((T(1,ISLICE,LJ)-T(1,ISLICE,LOUT)) +
Z (T(1,ISLICE,LJ)-T(1,ISLICE,L)))*TCOF(LJ,13)/TREPS)

C
J1 = 12 - L + LUP
J2 = 12 - L + LDN

C
THETA1 = (DX1+DX2+DX5+DX6)/(DX1+DX2+DX9+DX10)
THETA2 = ((TAU(L)+TAU(LUP))/(2.*DX1))*2.*TAU(L)/(DX1+DX2+DX9+DX10)
THETA3 = ((TAU(L)+TAU(LDN))/(2.*DX2))*2.*TAU(L)/(DX1+DX2+DX9+DX10)
THETA6 = 24.*TAU(L)/(AKW(ISLICE,IS)*S(ISLICE)*(DX1+DX2+DX9+DX10))
THETA4 = 0.0
THETA5 = 0.0
HUB1 = 0.0
HUB3 = 0.0
TIP1 = 0.0
TIP3 = 0.0
IF (ISLICE.EQ.1) GO TO 3290

C
C FOR A SLICE THAT IS NOT AT THE HUB OF THE BLADE:
C
THETA4 = (TAU(L)/S(ISLICE))*(TAU(L)/(S(ISLICE)+S(ISLICE-1)))*

```

```

NTARAYT 4441
NTARAYT 4442
NTARAYT 4443
NTARAYT 4444
NTARAYT 4445
NTARAYT 4446
NTARAYT 4447
NTARAYT 4448
NTARAYT 4449
NTARAYT 4450
NTARAYT 4451
NTARAYT 4452
NTARAYT 4453
NTARAYT 4454
NTARAYT 4455
NTARAYT 4456
NTARAYT 4457
NTARAYT 4458
NTARAYT 4459
NTARAYT 4460
NTARAYT 4461
NTARAYT 4462
NTARAYT 4463
NTARAYT 4464
NTARAYT 4465
NTARAYT 4466
NTARAYT 4467
NTARAYT 4468
NTARAYT 4469
NTARAYT 4470
NTARAYT 4471
NTARAYT 4472
NTARAYT 4473
NTARAYT 4474
NTARAYT 4475
NTARAYT 4476
NTARAYT 4477
NTARAYT 4478
NTARAYT 4479
NTARAYT 4480
NTARAYT 4481
NTARAYT 4482
NTARAYT 4483
NTARAYT 4484
NTARAYT 4485
NTARAYT 4486
NTARAYT 4487
NTARAYT 4488
NTARAYT 4489
NTARAYT 4490
NTARAYT 4491
NTARAYT 4492
NTARAYT 4493
NTARAYT 4494
NTARAYT 4495
NTARAYT 4496
NTARAYT 4497
NTARAYT 4498
NTARAYT 4499
NTARAYT 4500

```

```

      Z      2.*(DX1+DX2)/(DX1+DX2+DX9+DX10)      NTARAYT 4501
      IF (ISLICE.EQ.NSLICE) GO TO 3292      NTARAYT 4502
      THETA5 = THETA4*(S(ISLICE)+S(ISLICE-1))/(S(ISLICE)+S(ISLICE+1))      NTARAYT 4503
      TBELOW = T(1,ISLICE-1,L)      NTARAYT 4504
      TABOVE = T(1,ISLICE+1,L)      NTARAYT 4505
      GO TO 3294      NTARAYT 4506
C      NTARAYT 4507
C FOR THE SLICE AT THE HUB END OF THE BLADE:      NTARAYT 4508
C      NTARAYT 4509
3290 CONTINUE      NTARAYT 4510
      IF (IHUB.EQ.1) HUB1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*      NTARAYT 4511
      Z      (2.*(TAU(L)/S(1))**2)      NTARAYT 4512
C FOR IHUB = 1, HUB TEMPERATURE IS SPECIFIED*****      NTARAYT 4513
      THETA5 = 0.0      NTARAYT 4514
      IF (NSLICE.GT.1) THETA5 = (TAU(L)/S(1))*(TAU(L)/(S(1)+S(2)))*      NTARAYT 4515
      Z      (2.*(DX1+DX2)/(DX1+DX2+DX9+DX10))      NTARAYT 4516
      IF (IHUB.EQ.3) HUB3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/      NTARAYT 4517
      Z      (AKW(1,IS)*12.*S(1))      NTARAYT 4518
C IHUB = 3, THE HEAT FLUX AT THE HUB END IS SPECIFIED (BTU/HR FT**2) ***      NTARAYT 4519
      TBELOW = T(1,1,L)      NTARAYT 4520
      IF (IHUB.EQ.1) TBELOW = THUB(IS)      NTARAYT 4521
      TABOVE = T(1,1,L)      NTARAYT 4522
      IF (NSLICE.GT.1) TABOVE = T(1,2,L)      NTARAYT 4523
      IF (NSLICE.GT.1) GO TO 3294      NTARAYT 4524
C      NTARAYT 4525
C FOR THE SLICE AT THE BLADE TIP, (IF THERE ARE MORE THAN 1 SLICES      NTARAYT 4526
C BEING CONSIDERED) : *****      NTARAYT 4527
C      NTARAYT 4528
3292 CONTINUE      NTARAYT 4529
      IF (ITIP.EQ.1) TIP1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*      NTARAYT 4530
      Z      (2.*(TAU(L)/S(NSLICE))**2)      NTARAYT 4531
      IF (NSLICE.GT.1) TBELOW = T(1,ISLICE-1,L)      NTARAYT 4532
      IF (ITIP.EQ.3) TIP3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/      NTARAYT 4533
      Z      (AKW(NSLICE,IS)*12.*S(NSLICE))      NTARAYT 4534
      TABOVE = T(1,ISLICE,L)      NTARAYT 4535
      IF (ITIP.EQ.1) TABOVE = TTIP(IS)      NTARAYT 4536
C      NTARAYT 4537
C      NTARAYT 4538
3294 CONTINUE      NTARAYT 4539
      THETA9 = 0.0      NTARAYT 4540
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) THETA9 = 2.*3600.*RHOM*SPHTM*      NTARAYT 4541
      Z      (DX1+DX2)*(TAU(L)**2)/(144.*AKW(ISLICE,IS)*      NTARAYT 4542
      Z      (DX1+DX2+DX9+DX10)*DLTYME)      NTARAYT 4543
C      NTARAYT 4544
      TCOF(L,11) = 1.0*TREPS      NTARAYT 4545
      TCOF(L,13) = THETA1*TREPS      NTARAYT 4546
      TCOF(L,J1) = THETA2*TREPS      NTARAYT 4547
      TCOF(L,J2) = THETA3*TREPS      NTARAYT 4548
      TCOF(L,12) = (-1.0 - THETA1 - THETA2 - THETA3 - THETA4 - THETA5 -      NTARAYT 4549
      Z      HUB1 - TIP1)*TREPS - THETA9      NTARAYT 4550
      TCOF(L,24) = QSNK(IS)*THETA6 - (THETA4+HUB1)*TBELOW      NTARAYT 4551
      Z      -(THETA5+TIP1)*TABOVE - QHUB(IS)*HUB3 + QTIP(IS)*TIP3      NTARAYT 4552
      Z      -(1-TREPS)*(T(1,ISLICE,LUP)*THETA2+T(1,ISLICE,LJ)      NTARAYT 4553
      Z      +T(1,ISLICE,LIN)*THETA1+T(1,ISLICE,LDN)*THETA3)      NTARAYT 4554
      Z      +T(1,ISLICE,L)*((1.0-TREPS)*(1.+THETA1+THETA2+THETA3      NTARAYT 4555
      Z      +THETA4+THETA5+HUB1+TIP1) - THETA9)      NTARAYT 4556
C      NTARAYT 4557
C      NTARAYT 4558
      AHTRN1 = (DX5 + DX6)*S(ISLICE)/2.      NTARAYT 4559
      THETA8 = 2.*HX*HC(IS)*AHTRN1*TAU(L)/(12.*AKW(ISLICE,IS)*S(ISLICE)      NTARAYT 4560

```

Z	*(DX1+DX2+DX5+DX6))	NTARAYT 4561
	TCOF(LIN,11) = TREPS	NTARAYT 4562
	TCOF(LIN,24) = - THETA8*T0G - (1.0-TREPS)*(T(1,ISLICE,L)	NTARAYT 4563
Z	-T(1,ISLICE,LIN))*(1.0+THETA8))	NTARAYT 4564
	TCOF(LIN,12) = (-1.0 - THETA8)*TREPS	NTARAYT 4565
C		NTARAYT 4566
	TCOF(LCOOL,12) = 1.0	NTARAYT 4567
	TCOF(LCOOL,24) = T0G	NTARAYT 4568
	IF (TYME.GT.0.) RCHRD = (144./3600.)*AKW(ISLICE,IS)*DLTYME/	NTARAYT 4569
Z	(RHOM*SPHTM*((DX1+DX2)/2.))**2)	NTARAYT 4570
	IF (TYME.GT.0.) RTRNV = (144./3600.)*AKW(ISLICE,IS)*DLTYME/	NTARAYT 4571
Z	(RHOM*SPHTM*(TAU(L)**2))	NTARAYT 4572
	IF (RCHRD.GT.RCHRDM) RCHRDM = RCHRD	NTARAYT 4573
	IF (RTRNV.GT.RTRNVN) RTRNVN = RTRNV	NTARAYT 4574
C		NTARAYT 4575
	GO TO 440	NTARAYT 4576
C		NTARAYT 4577
C		NTARAYT 4578
C		NTARAYT 4579
C		NTARAYT 4580
C		NTARAYT 4581
C		NTARAYT 4582
C		NTARAYT 4583
C		NTARAYT 4584
C		NTARAYT 4585
C	*****	NTARAYT 4586
C	*****	NTARAYT 4587
C	***** THIS BLOCK COMPUTES TCOF ELEMENTS FOR GENERAL FLOW STATIONS	NTARAYT 4588
C	*****	NTARAYT 4589
C	*****	NTARAYT 4590
C		NTARAYT 4591
330	CONTINUE	NTARAYT 4592
	DX1 = DLX(L)	NTARAYT 4593
	DX2 = DLX(LDN)	NTARAYT 4594
	DX3 = DLX(LOUT)	NTARAYT 4595
	DX4 = DLX(LDN-2)	NTARAYT 4596
	DX5 = DLX(LIN)	NTARAYT 4597
	DX6 = DLX(LDN+1)	NTARAYT 4598
	DX7 = DLX(LCOOL)	NTARAYT 4599
	DX9 = DLX(LJ)	NTARAYT 4600
	DX10 = DLX(LDN-1)	NTARAYT 4601
	IF (IS.GT.1) GO TO 332	NTARAYT 4602
C		NTARAYT 4603
	IF THIS IS STATION NUMBER 1:	NTARAYT 4604
	DX1 = DLX(LUP)	NTARAYT 4605
	DX9 = DLX(LUP-1)	NTARAYT 4606
	DX3 = DLX(LUP-2)	NTARAYT 4607
	DX5 = DLX(LUP+1)	NTARAYT 4608
	DX7 = DLX(LUP+2)	NTARAYT 4609
C		NTARAYT 4610
332	CONTINUE	NTARAYT 4611
	LCUPP = LCOOL - 5	NTARAYT 4612
	LCOOLP = LCOOL + 4	NTARAYT 4613
	J1 = 12 - L + LUP	NTARAYT 4614
	J2 = 12 - L + LDN	NTARAYT 4615
	J4 = 12 - LCOOL + LCUPS	NTARAYT 4616
	J5 = 12 - LCOOL + LCUP	NTARAYT 4617
	J6 = 12 - LCOOL + LCUPP	NTARAYT 4618
	J8 = 12 - LCOOL + LCOOLP	NTARAYT 4619
C		NTARAYT 4620
333	CONTINUE	

```

      AHTRN1 = (DX5 + DX6)*S(ISLICE)/2.
      A1 = SPAN*DX5/2.
      A2 = A1
      A3 = 0.0
      A4 = 0.0
C
      IF (IHC(IS).EQ.3.AND.IS.LE.NPWD)
Z      CALL HCPINS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EPAREA)
      IF (IHC(IS-2).EQ.3) A1 = EPAREA (IS-2)/2.
      IF (IHC(IS).EQ.3) A2 = EPAREA (IS)/2.
      IF (IHC(IS).EQ.3) AHTRN1 = EPAREA (IS)
C
      I3 = 12 - LIN + LCOOL
340  CONTINUE
C
      CURV = 1.0 + (DX9+DX10)/(DX3+DX4)
C
C*****
C
C FOR MID-METAL NODE (L) :
C      TCOF(L,12), 12 REFERS TO NODE L
C      (L,J1), J1 REFERS TO NODE LUP
C      (L,J2), J2 REFERS TO NODE LDN
C      (L,11), 11 REFERS TO NODE LJ
C      (L,13), 13 REFERS TO NODE LIN
C FOR THE COOLANT NODE (LCOOL) :
C      TCOF(LCOOL,12), 12 REFERS TO NODE LCOOL
C      (LCOOL,J4), J4 REFERS TO NODE LCUPS = LUP+1
C      ,J5), J5 REFERS TO NODE LCUP
C      ,J6), J6 REFERS TO NODE LCUPP, (TRAILING EDGE REGION ONLY)
C      ,11), 11 REFERS TO NODE LIN
C      ,J8), J8 REFERS TO NODE LCOOLP, (TRAILING EDGE REGION ONLY)
C*****
C
C FOR OUTSIDE NODE:
C
      TRTRMC = 0.0
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMC =
Z (3600./144.)*RHOC*SPHTC*(TAU(LOUT)**2)/(4.*AKC (ISLICE,IS) *DLTYME)
      TCOF(LOUT,13) = -TREPS + TRTRMC*CURV
      TCOF(LOUT,12) = (1.0 + (1.0-BTA)*HG (IS)*TAU(LOUT)/
Z (12.*AKC (ISLICE,IS)))*TREPS + TRTRMC*CURV
      TCOF(LOUT,J9) = -(1.-BTA)*TREPS*FLMEFF (IS)*HG (IS)*TAU(LOUT)/
Z (12.*AKC (ISLICE,IS))
      TCOF(LOUT,24) = (BTA*QG (IS) + (1.0-BTA)*HG (IS)*TQ (IS)*
Z (1.0-FLMEFF (IS)))*TAU(LOUT)/(12.*AKC (ISLICE,IS))
      Z - T(1,ISLICE,LOUT)*((1.-TREPS)*((1.-BTA)*HG (IS)*TAU(LOUT)/
      Z (12.*AKC (ISLICE,IS)) + 1.) - TRTRMC*CURV)
      Z + T(1,ISLICE,LJ)*(1.-TREPS+TRTRMC*CURV)
      Z + T(1,ISLICE,LCOOL)*(1.-TREPS)*FLMEFF (IS)*(1.-BTA)*HG (IS)*
      Z TAU(LOUT)/(12.*AKC (ISLICE,IS))
C
C
C AT JUNCTION OF COATING AND METAL, NODE LJ:
C
      TCOF(LJ,11) = TREPS
      TCOF(LJ,13) = TREPS*(AKW (ISLICE,IS)/AKC (ISLICE,IS))*
Z (2.*TAU (LOUT)/TAU (L))*(DX1+DX2+DX3+DX4)/(DX9+DX10+DX3+DX4)
      TCOF(LJ,12) = - TCOF(LJ,11) - TCOF(LJ,13)

```

```

NTARAYT 4621
NTARAYT 4622
NTARAYT 4623
NTARAYT 4624
NTARAYT 4625
NTARAYT 4626
NTARAYT 4627
NTARAYT 4628
NTARAYT 4629
NTARAYT 4630
NTARAYT 4631
NTARAYT 4632
NTARAYT 4633
NTARAYT 4634
NTARAYT 4635
NTARAYT 4636
NTARAYT 4637
NTARAYT 4638
NTARAYT 4639
NTARAYT 4640
NTARAYT 4641
NTARAYT 4642
NTARAYT 4643
NTARAYT 4644
NTARAYT 4645
NTARAYT 4646
NTARAYT 4647
NTARAYT 4648
NTARAYT 4649
NTARAYT 4650
NTARAYT 4651
NTARAYT 4652
NTARAYT 4653
NTARAYT 4654
NTARAYT 4655
NTARAYT 4656
NTARAYT 4657
NTARAYT 4658
NTARAYT 4659
NTARAYT 4660
NTARAYT 4661
NTARAYT 4662
NTARAYT 4663
NTARAYT 4664
NTARAYT 4665
NTARAYT 4666
NTARAYT 4667
NTARAYT 4668
NTARAYT 4669
NTARAYT 4670
NTARAYT 4671
NTARAYT 4672
NTARAYT 4673
NTARAYT 4674
NTARAYT 4675
NTARAYT 4676
NTARAYT 4677
NTARAYT 4678
NTARAYT 4679
NTARAYT 4680

```



```

TCOP(LJ,24) = (1.-TREPS)*((T(1,ISLICE,LJ)-T(1,ISLICE,LOUT)) + NTARAYT 4681
Z      (T(1,ISLICE,LJ)-T(1,ISLICE,L)))*TCOP(LJ,13)/TREPS) NTARAYT 4682
C NTARAYT 4683
C NTARAYT 4684
C FOR MID-METAL NODE: NTARAYT 4685
C NTARAYT 4686
C THETA1 = (DX1+DX2+DX5+DX6)/(DX1+DX2+DX9+DX10) NTARAYT 4687
C THETA2 = ((TAU(L)+TAU(LUP))/(2.*DX1))*2.*TAU(L)/(DX1+DX2+DX9+DX10) NTARAYT 4688
C THETA3 = 0.0 NTARAYT 4689
C THETA6 = 24.*TAU(L)/(AKW(ISLICE,IS)*S(ISLICE)*(DX1+DX2+DX9+DX10)) NTARAYT 4690
C IF (IS.LT.NSTA-1) THETA3 = ((TAU(L)+TAU(LDN))/(2.*DX2))*2.* NTARAYT 4691
Z      TAU(L)/(DX1+DX2+DX9+DX10) NTARAYT 4692
C THETA4 = 0.0 NTARAYT 4693
C THETA5 = 0.0 NTARAYT 4694
C HUB1 = 0.0 NTARAYT 4695
C HUB3 = 0.0 NTARAYT 4696
C TIP1 = 0.0 NTARAYT 4697
C TIP3 = 0.0 NTARAYT 4698
C IF (ISLICE.EQ.1) GO TO 3410 NTARAYT 4699
C NTARAYT 4700
C FOR A SLICE THAT IS NOT AT THE HUB OF THE BLADE: NTARAYT 4701
C NTARAYT 4702
C THETA4 = (TAU(L)/S(ISLICE))*(TAU(L)/(S(ISLICE)+S(ISLICE-1)))* NTARAYT 4703
Z      2.*(DX1+DX2)/(DX1+DX2+DX9+DX10) NTARAYT 4704
C IF (ISLICE.EQ.NSLICE) GO TO 3412 NTARAYT 4705
C THETA5 = THETA4*(S(ISLICE)+S(ISLICE-1))/(S(ISLICE)+S(ISLICE+1)) NTARAYT 4706
C TBELOW = T(1,ISLICE-1,L) NTARAYT 4707
C TABOVE = T(1,ISLICE+1,L) NTARAYT 4708
C GO TO 3414 NTARAYT 4709
C NTARAYT 4710
C FOR THE SLICE AT THE HUB END OF THE BLADE: NTARAYT 4711
C NTARAYT 4712
C 3410 CONTINUE NTARAYT 4713
C IF (IHUB.EQ.1) HUB1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(2.*(TAU(L)/ NTARAYT 4714
Z      S(1))*2) NTARAYT 4715
C FOR IHUB = 1, HUB TEMPERATURE IS SPECIFIED***** NTARAYT 4716
C NTARAYT 4717
C THETA5 = 0.0 NTARAYT 4718
C IF (NSLICE.GT.1) THETA5 = (TAU(L)/S(1))*(TAU(L)/(S(1)+S(2)))* NTARAYT 4719
Z      (2.*(DX1+DX2)/(DX1+DX2+DX9+DX10)) NTARAYT 4720
C NTARAYT 4721
C IF (IHUB.EQ.3) HUB3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/ NTARAYT 4722
Z      (AKW(1,IS)*12.*S(1)) NTARAYT 4723
C IHUB = 3, THE HEAT FLUX AT THE HUB END IS SPECIFIED (BTU/HR FT**2) *** NTARAYT 4724
C NTARAYT 4725
C TBELOW = T(1,1,L) NTARAYT 4726
C IF (IHUB.EQ.1) TBELOW = THUB(IS) NTARAYT 4727
C NTARAYT 4728
C TABOVE = T(1,1,L) NTARAYT 4729
C IF (NSLICE.GT.1) TABOVE = T(1,2,L) NTARAYT 4730
C NTARAYT 4731
C IF (NSLICE.GT.1) GO TO 3414 NTARAYT 4732
C NTARAYT 4733
C FOR THE SLICE AT THE BLADE TIP, (IF THERE ARE MORE THAN 1 SLICES NTARAYT 4734
C BEING CONSIDERED) : ***** NTARAYT 4735
C NTARAYT 4736
C 3412 CONTINUE NTARAYT 4737
C IF (ITIP.EQ.1) TIP1 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(2.*(TAU(L)/ NTARAYT 4738
Z      S(NSLICE))*2) NTARAYT 4739
C IF (NSLICE.GT.1) TBELOW = T(1,ISLICE-1,L) NTARAYT 4740

```

```

      IF (ITIP.EQ.3) TIP3 = ((DX1+DX2)/(DX1+DX2+DX9+DX10))*(TAU(L)**2)/
Z      (AKW (NSLICE, IS) *12.*S (NSLICE))
      TABOVE = T(1, ISLICE, L)
      IF (ITIP.EQ.1) TABOVE = TTIP(IS)
C
3414 CONTINUE
      THETA9 = 0.0
      IF (DLTYME.GT.0.0.AND.TYME.GE.0.) THETA9 = 2.*3600.*RHOM*SPHTM*
Z      (DX1+DX2)*(TAU(L)**2)/(144.*AKW (ISLICE, IS) *
Z      (DX1+DX2+DX9+DX10)*DLTYME)
C
C
C ENDEFF IS EFFECT OF HEAT TRANSFER FROM THE GAS TO THE REAR EDGE OF
C THE BLADE
C
      ENDEFF = 0.0
      ENDFLX = 0.0
      IF (IS.GE.NSTA-1.AND.BTA.EQ.0.0) ENDEFF = 2.*HG (IS) *(TAU(L)**2)/
Z      (12.*AKW (ISLICE, IS) *(DX1+DX9))
      IF (IS.GE.NSTA-1.AND.BTA.GT.0.0) ENDFLX = QG (IS) *(TAU(L)**2)/
Z      (12.*AKW (ISLICE, IS) *(DX1+DX9))
3416 CONTINUE
      TCOP(L,11) = 1.0*TREPS
      TCOP(L,13) = THETA1*TREPS
      TCOP(L,J1) = THETA2*TREPS
      TCOP(L,J2) = 0.0
      TCOP(L,12) = (-1.0 - THETA1 - THETA2 - THETA3 - THETA4 - THETA5
Z      - ENDEFF - HUB1 - TIP1)*TREPS - THETA9
      TCOP(L,24) = QSNK(IS)*THETA6 - (THETA4+HUB1)*TBELOW -
Z      (THETA5+TIP1)*TABOVE - QHUB(IS)*HUB3 + QTIP(IS)*TIP3
Z      - (1.-TREPS)*(T(1, ISLICE, LUP)*THETA2+T(1, ISLICE, LJ) +
Z      T(1, ISLICE, LIN)*THETA1+T(1, ISLICE, LDN)*THETA3)
Z      +T(1, ISLICE, L)*((1.0-TREPS)*(1.+THETA1+THETA2+THETA3+
Z      THETA4+THETA5+HUB1+TIP1+ENDEFF)
Z      - THETA9) - TG (IS) *ENDEFF - ENDFLX
      IF (IS.LT.NSTA-1) TCOP(L,J2) = THETA3*TREPS
C
      PUMP (IS) = (.1047198*WS)**2*RR (IS) *(RR (IS) -RR (IS-2) )
C
C FOR INNER SURFACE NODE:
C
342 THETA8 = 2.*HX*HC (IS) *AHTRN1*TAU (L) /(12.*AKW (ISLICE, IS) *
Z      S (ISLICE) *(DX1+DX2+DX5+DX6) )
C
      TCOP (LIN,11) = TREPS
      TCOP (LIN,12) = (-1.0 - THETA8)*TREPS
      TCOP (LIN,13) = THETA8*(YCONV + YFINS) *TREPS
      TCOP (LIN,24) = - YIMP*THETA8*T0G
Z      - (1.-TREPS)*(T(1, ISLICE, L)-T(1, ISLICE, LIN) *(1.+THETA8)
Z      +THETA8*(YCONV+YFINS)*T(1, ISLICE, LCOOL) )
C
C IF THIS IS A TRAILING EDGE, PRESSURE SIDE, STATION, COOLANT NODE
C COINCIDES WITH SUCTION SIDE COOLANT NODE.
      IF (ISENS.EQ.0) GO TO 343
      TCOP(LCOOL,7) = -1.0
      TCOP(LCOOL,12) = 1.0
      TCOP(LCOOL,24) = 0.0
      IF (ISENS.EQ.1) GO TO 430
343 CONTINUE
C

```

```

NTARAYT 4741
NTARAYT 4742
NTARAYT 4743
NTARAYT 4744
NTARAYT 4745
NTARAYT 4746
NTARAYT 4747
NTARAYT 4748
NTARAYT 4749
NTARAYT 4750
NTARAYT 4751
NTARAYT 4752
NTARAYT 4753
NTARAYT 4754
NTARAYT 4755
NTARAYT 4756
NTARAYT 4757
NTARAYT 4758
NTARAYT 4759
NTARAYT 4760
NTARAYT 4761
NTARAYT 4762
NTARAYT 4763
NTARAYT 4764
NTARAYT 4765
NTARAYT 4766
NTARAYT 4767
NTARAYT 4769
NTARAYT 4769
NTARAYT 4770
NTARAYT 4771
NTARAYT 4772
NTARAYT 4773
NTARAYT 4774
NTARAYT 4775
NTARAYT 4776
NTARAYT 4777
NTARAYT 4778
NTARAYT 4779
NTARAYT 4780
NTARAYT 4781
NTARAYT 4782
NTARAYT 4783
NTARAYT 4784
NTARAYT 4785
NTARAYT 4786
NTARAYT 4787
NTARAYT 4788
NTARAYT 4789
NTARAYT 4790
NTARAYT 4791
NTARAYT 4792
NTARAYT 4793
NTARAYT 4794
NTARAYT 4795
NTARAYT 4796
NTARAYT 4797
NTARAYT 4798
NTARAYT 4799
NTARAYT 4800

```

```

C FOR COOLANT NODE:
C FOR THE SPECIAL CASE OF IS = NFWD+1, GO TO 350
C
FILMW = WFC (IS)
IF (IS.GT.NFWD) FILMW = FILMW + WFC (IS+1)
IF (IS.EQ.NFWD+1) GO TO 350
C
C
WXCP = WCROS (2, ISLICE, IS) * 144. * CPC (IS) * 3600.
IF (IS.EQ.ICOMP.OR.IS.EQ.ICOMS) WXCP = WJ (ISLICE, JS) * 144. *
Z CPC (IS) * 3600. / 2.
C
C
C DEFINE A COOLANT SIDE TRANSIENT TERM, TRTRMG
C
TRTRMG = 0.0
IF (DLTYME.GT.0.0.AND.TYME.GE.0.) TRTRMG = (1.+CPC (IS-2) / CPC (IS)) *
Z (P (1, ISLICE, IS-2) / T (1, ISLICE, LCUP)
Z - P (1, ISLICE, IS) / T (1, ISLICE, LCOOL))
Z * (A (LCUP) + A (LCOOL)) * DLX (LCOOL) / (16. * DLTYME * F*
Z WCROS (2, ISLICE, IS) * 12.)
TCOF (LCOOL, J4) = HX * TREPS * HC (ISUP) * A1 / WXCP
TCOF (LCOOL, J5) = 144. * 3600. * TREPS * WCROS (2, ISLICE, ISUP) *
Z (1.+AM2 (ISUP) * (GAMC (ISUP) - 1.) / 2.) * CPC (ISUP) / WXCP
Z - HX * TREPS * HC (ISUP) * (YCONVU + YFINSU) * (A1+A3) / WXCP - TRTRMG
TCOF (LCOOL, 11) = HX * TREPS * HC (IS) * A2 / WXCP
TCOF (LCOOL, 12) = TREPS * ((-FILMW * 144. * 3600. * CPC (IS) / WXCP) - 1.0
Z - ((GAMC (IS) - 1.) / 2.) * AM2 (IS)
Z - HX * HC (IS) * (A2 + A4) * (YCONV + YFINS) / WXCP - TRTRMG
TREDGE = 0.0
IF (IS.GT.NFWD) TREDGE = ((1.-TREPS) * HX / WXCP) * (T (1, ISLICE, LCUP) *
Z HC (ISUP) * A3 + T (1, ISLICE, LCOOL) * HC (IS) * A4)
TCOF (LCOOL, 24) = -(CPO * FACTOR * WJ (ISLICE, ISUP) * 144. * 3600. / WXCP) * TOG
Z - PUMP (IS) / (CPC (IS) * 778. * 144. * 32.2)
Z + TOG * (HX * HC (ISUP) * A1 * YIMPU + HX * HC (IS) * A2 * YIMP) / WXCP
Z - T (1, ISLICE, LCUP) * HX * (1.-TREPS) * HC (ISUP) * A1 / WXCP
Z - T (1, ISLICE, LCUP) * (144. * 3600. * (1.-TREPS) * WCROS (2, ISLICE, ISUP)
Z * (1.+AM2 (ISUP) * (GAMC (ISUP) - 1.) / 2.) * CPC (ISUP) /
Z WXCP - HX * (1.-TREPS) * HC (ISUP) * (YCONVU +
Z YFINSU) * (A1+A3) / WXCP - TRTRMG)
Z - T (1, ISLICE, LIN) * HX * (1.-TREPS) * HC (IS) * A2 / WXCP
Z - T (1, ISLICE, LCOOL) * ((1.-TREPS) * ((-FILMW * 144. * 3600. * CPC (IS) /
Z WXCP) - 1.0 - ((GAMC (IS) - 1.) / 2.) * AM2 (IS)
Z - HX * HC (IS) * (A2 + A4) * (YCONV + YFINS) / WXCP - TRTRMG) - TREDGE
IF (IS.GT.NFWD) TCOF (LCOOL, J6) = TREPS * HX * HC (ISUP) * A3 / WXCP
IF (IS.GT.NFWD) TCOF (LCOOL, J8) = TREPS * HX * HC (IS) * A4 / WXCP
C
C*** OF THE TERMS YIMP, YFINS, YCONV, ONLY ONE CAN BE NON-ZERO AT A TIME
C YIMP = 1.0 MEANS THAT IMPINGEMENT HEAT TRANSFER IS BEING CONSIDERED
C YFINS = 1.0 MEANS THAT A PIN FINNED SURFACE IS BEING USED
C YCONV = 1.0 MEANS A FORCED CONVECTION CORRELATION IS BEING USED
C
C
IF (TYME.GT.0.) RCHRD = (144. / 3600.) * AKW (ISLICE, IS) * DLTYME /
Z (RHOM * SPHTM * (((DX1+DX2) / 2.) ** 2))
IF (TYME.GT.0.) RTRNV = (144. / 3600.) * AKW (ISLICE, IS) * DLTYME /
Z (RHOM * SPHTM * (TAU (L) ** 2))
IF (RCHRD.GT.RCHRD) RCHRD = RCHRD
IF (RTRNV.GT.RTRNV) RTRNV = RTRNV
GO TO 430

```

```

NTARAYT 4801
NTARAYT 4802
NTARAYT 4803
NTARAYT 4804
NTARAYT 4805
NTARAYT 4806
NTARAYT 4807
NTARAYT 4808
NTARAYT 4809
NTARAYT 4810
NTARAYT 4811
NTARAYT 4812
NTARAYT 4813
NTARAYT 4814
NTARAYT 4815
NTARAYT 4816
NTARAYT 4817
NTARAYT 4818
NTARAYT 4819
NTARAYT 4820
NTARAYT 4821
NTARAYT 4822
NTARAYT 4823
NTARAYT 4824
NTARAYT 4825
NTARAYT 4826
NTARAYT 4827
NTARAYT 4828
NTARAYT 4829
NTARAYT 4830
NTARAYT 4831
NTARAYT 4832
NTARAYT 4833
NTARAYT 4834
NTARAYT 4835
NTARAYT 4836
NTARAYT 4837
NTARAYT 4838
NTARAYT 4839
NTARAYT 4840
NTARAYT 4841
NTARAYT 4842
NTARAYT 4843
NTARAYT 4844
NTARAYT 4845
NTARAYT 4846
NTARAYT 4847
NTARAYT 4848
NTARAYT 4849
NTARAYT 4850
NTARAYT 4851
NTARAYT 4852
NTARAYT 4853
NTARAYT 4854
NTARAYT 4855
NTARAYT 4856
NTARAYT 4857
NTARAYT 4858
NTARAYT 4859
NTARAYT 4860

```



```

350 CONTINUE
C
C FOR MIXING ZONE; STATION NO. NFW+1:
C
  IF (IHC(IS).EQ.3) CALL HCPINS (IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS
Z      ,EFAREA)
  AHTRN1 = EFAREA (IS)
  IF (IHC(IS).EQ.3) A2 = EFAREA (IS) /2.
  A3 = SPAN*DLX (LCOOLP) /2.
  IF (IHC (IS-1) .EQ.3) A3 = EFAREA (IS-1) /2.
  A4 = SPAN*DLX (LCOOLP) /2.
  IF (IHC (IS) .EQ.3) A4 = EFAREA (IS+1) /2.
  IF (IHC (IS) .EQ.3) GO TO 360
C
  IF (IHC (IS) .EQ.2) HC (IS) = HCFRCD (IS,LCOOL,LIN)
  AHTRN1 = (DX5+DX6)*S (ISLICE) /2.
360 CONTINUE
C
  IF (IHC (IS-1) .EQ.1) YIMPUU = 1.0
  IF (IHC (IS-1) .EQ.2) YCNVUU=1.0+RCVRY*AM2 (IS-1) * (GAMC (IS-1) -1.) /2.
  IF (IHC (IS-1) .EQ.3) YFNSUU = 1.0
C
  WXCP = WCROS (2,ISLICE,IS)*144.*CPC (IS)*3600.
  RHOBAR = ((P(1,ISLICE,IS-2)+P(1,ISLICE,IS))/24.)*(1./R)*2.*
Z      WCROS (2,ISLICE,IS) /
Z      (T(1,ISLICE,LCUPP+1)*WCROS (2,ISLICE,IS-1) + T(1,ISLICE,LCUP) *
Z      WCROS (2,ISLICE,IS-2)
Z      + TOG*WDUMP + T(1,ISLICE,LCOOL)*WCROS (2,ISLICE,IS))
  VOLBAR = (A (LCUP) +A (LCUPP+1) +A (LCOOL) ) * (DLX (LCOOLP) +DLX (LIN) ) /4.
  TRTRMJ = 0.0
  IF (DLTIME.GT.0.0.AND.TYME.GE.0.) TRTRMJ = PHOBAR*VOLBAR/
Z      (2.*DLTIME*(WCROS (2,ISLICE,IS)**2))
  TCOF (LCOOL,1) = TREPS*HX*HC (IS-2) *A1/WXCP
  TCOF (LCOOL,2) = TREPS*( (WCROS (2,ISLICE,ISUP) /WCROS (2,ISLICE,IS) )
Z      *(1.+AM2 (ISUP) * (GAMC (ISUP) -1.) /2.) * (CPC (ISUP) /
Z      CPC (IS) ) - HX*HC (ISUP) * (YCNVUU+YFNSUU) * (A1) /WXCP)
Z      - TRTRMJ*WCROS (2,ISLICE,IS-2)
  TCOF (LCOOL,6) = TREPS*HX*HC (IS-1) *A3/WXCP
  TCOF (LCOOL,7) = TREPS*( (WCROS (2,ISLICE,IS-1) /WCROS (2,ISLICE,IS) ) *
Z      (1.+AM2 (IS-1) * (GAMC (IS-1) -1.) /2.) * (CPC (IS-1) /
Z      CPC (IS) ) - HX*HC (IS-1) * (YCNVUU+YFNSUU) * (A3) /WXCP)
Z      - TRTRMJ*WCROS (2,ISLICE,IS-1)
  TCOF (LCOOL,11) = TREPS*HX*HC (IS) *A2/WXCP
  TCOF (LCOOL,12) = TREPS*( (-FILMW/WCROS (2,ISLICE,IS) ) -1.0- ( (GAMC (IS)
Z      -1.) /2.) *AM2 (IS) -HX*HC (IS) * (A2+A4) * (YCONV+YFINS)
Z      /WXCP) - TRTRMJ*WCROS (2,ISLICE,IS)
  TCOF (LCOOL,16) = TREPS*HX*HC (IS) *A4/WXCP
  TCOF (LCOOL,24) = TOG*(HX*HC (IS-2) *A1*YIMPU + HX*HC (IS) * (A2+A4) *
Z      YIMP + HX*HC (IS-1) *A3*YIMPUU) /WXCP
Z      - PUMP (IS) / (CPC (IS) *778.*144.*32.2) - (CPO/CPC (IS) ) *TOG*
Z      (WJ (ISLICE,IS-2) +WJ (ISLICE,IS-1) +WDUMP) /WCROS (2,ISLICE,IS)
Z      -T (1,ISLICE,LCUPS) * (1.-TREPS) *HX*HC (IS-2) *A1/WXCP
Z      -T (1,ISLICE,LCUP) * ( (1.-TREPS) * ( (WCROS (2,ISLICE,ISUP) /
Z      WCROS (2,ISLICE,IS) ) * (1.+AM2 (ISUP) * (GAMC (ISUP) -1.) /2.)
Z      * (CPC (ISUP) /CPC (IS) ) - HX*HC (ISUP) * (YCONVU+YFINSU)
Z      * (A1) /WXCP) + TRTRMJ*WCROS (2,ISLICE,IS-2) )
Z      -T (1,ISLICE,LCUPP) * (1.-TREPS) *HX*HC (IS-1) *A3/WXCP
Z      -T (1,ISLICE,LCUPP+1) * ( (1.-TREPS) * ( (WCROS (2,ISLICE,IS-1) /
Z      WCROS (2,ISLICE,IS) ) * (1.+AM2 (IS-1) * (GAMC (IS-1) -1.) /2.)
Z      * (CPC (IS-1) /CPC (IS) ) - HX*HC (IS-1) * (YCNVUU+YFNSUU) * (A3) /

```


Z	WXCP) + TRTRMJ*WCROS(2,ISLICE,IS-1)	NTARAYT 4921
Z	-T(1,ISLICE,LIN)*(1.-TREPS)*HX*HC(IS)*A2/WXCP	NTARAYT 4922
Z	-T(1,ISLICE,LCOOL)*(1.-TREPS)*((-FILHW/WCROS(2,ISLICE,IS))-1.0	NTARAYT 4923
Z	-((GANC(IS)-1.)/2.)*AM2(IS)	NTARAYT 4924
Z	- HX*HC(IS)*(A2+A4)*(YCONV+YFINS)/WXCP)	NTARAYT 4925
Z	- T(1,ISLICE,LCOOL)*TRTRMJ*WCROS(2,ISLICE,IS)	NTARAYT 4926
Z	-T(1,ISLICE,LCOOLP)*(1.-TREPS)*HX*HC(IS)*A4/WXCP	NTARAYT 4927
C	GO TO 430	NTARAYT 4928
C		NTARAYT 4929
C		NTARAYT 4930
C		NTARAYT 4931
C		NTARAYT 4932
C		NTARAYT 4933
C		NTARAYT 4934
C		NTARAYT 4935
C		NTARAYT 4936
C		NTARAYT 4937
C		NTARAYT 4938
C	*****	NTARAYT 4939
C	*****	NTARAYT 4940
C	***** THIS BLOCK SETS UP TCOF ELEMENTS FOR STATIONS THAT ARE FORWARD	NTARAYT 4941
C	OF THE FLOW SPLIT POINT,	NTARAYT 4942
C	***** AND ARE ON THE SAME SIDE OF THE BLADE AS THE FLOW SPLIT POINT	NTARAYT 4943
C	*****	NTARAYT 4944
C	*****	NTARAYT 4945
C		NTARAYT 4946
C		NTARAYT 4947
370	CONTINUE	NTARAYT 4948
	ISUP = IS + 2	NTARAYT 4949
	LCUP = LCOOL + 10	NTARAYT 4950
	LCUPS = LCOOL + 9	NTARAYT 4951
	LUP = LCOOL + 8	NTARAYT 4952
	LDN = LCOOL - 12	NTARAYT 4953
	IF (IS.EQ.2) LDN = 3	NTARAYT 4954
	IF (IS.EQ.1) LDN = 8	NTARAYT 4955
	IF (IHC(ISUP).EQ.1) YIMPU = 1.0	NTARAYT 4956
	IF (IHC(ISUP).EQ.2) YCONVU=1.0+RCVRY*AM2(ISUP)*(GANC(ISUP)-1.)/2.	NTARAYT 4957
	IF (IHC(ISUP).EQ.3) YFINSU = 1.0	NTARAYT 4958
	DX1 = DLX(LUP)	NTARAYT 4959
	DX2 = DLX(L)	NTARAYT 4960
	DX3 = DLX(LUP-2)	NTARAYT 4961
	DX4 = DLX(LOUT)	NTARAYT 4962
	DX5 = DLX(LCUPS)	NTARAYT 4963
	DX6 = DLX(LIN)	NTARAYT 4964
	DX7 = DLX(LCUP)	NTARAYT 4965
	DX9 = DLX(LUP-1)	NTARAYT 4966
	DX10 = DLX(LJ)	NTARAYT 4967
	IF (IS.GT.1) GO TO 332	NTARAYT 4968
C		NTARAYT 4969
	DX2 = DLX(8)	NTARAYT 4970
	DX4 = DLX(6)	NTARAYT 4971
	DX6 = DLX(9)	NTARAYT 4972
	DX10 = DLX(7)	NTARAYT 4973
	J1 = 22	NTARAYT 4974
	J2 = 17	NTARAYT 4975
	J4 = 21	NTARAYT 4976
	J5 = 22	NTARAYT 4977
	J6 = 8	NTARAYT 4978
	J8 = 15	NTARAYT 4979
	GO TO 333	NTARAYT 4980
C		

C		NTARAYT 4981
C		NTARAYT 4982
C		NTARAYT 4983
C		NTARAYT 4984
C		NTARAYT 4985
C		NTARAYT 4986
C		NTARAYT 4987
C	*****	NTARAYT 4988
C	*****	NTARAYT 4989
C	***** SETUP FOR TRAILING EDGE REGION:	NTARAYT 4990
C	*****	NTARAYT 4991
C	*****	NTARAYT 4992
380	CONTINUE	NTARAYT 4993
	ISENS = IS - 2*(IS/2)	NTARAYT 4994
C		NTARAYT 4995
C	ISENS = 0 MEANS IS IS EVEN AND STATION IS ON SUCTION SIDE	NTARAYT 4996
C	ISENS = 1 MEANS IS IS ODD AND STATION IS ON PRESSURE SIDE	NTARAYT 4997
C		NTARAYT 4998
	LCOOL = 5*IS	NTARAYT 4999
	LIN = LCOOL - 1	NTARAYT 5000
	L = LCOOL - 2	NTARAYT 5001
	LJ = LCOOL - 3	NTARAYT 5002
	LOUT = LCOOL - 4	NTARAYT 5003
	LUP = L - 10	NTARAYT 5004
	LDN = L + 10	NTARAYT 5005
	LCUP = LCOOL - 10	NTARAYT 5006
	LCUPS = LCUP - 1	NTARAYT 5007
	LCUPP = LCOOL - 6	NTARAYT 5008
	LCOOLP = LCOOL + 4	NTARAYT 5009
C		NTARAYT 5010
C		NTARAYT 5011
	I3 = 12 - LIN + LCOOL	NTARAYT 5012
	J1 = 12 - L + LUP	NTARAYT 5013
	J2 = 12 - L + LDN	NTARAYT 5014
	J4 = 12 - LCOOL + LCUPS	NTARAYT 5015
	J5 = 12 - LCOOL + LCUP	NTARAYT 5016
	J6 = 12 - LCOOL + LCUPP	NTARAYT 5017
	J8 = 12 - LCOOL + LCOOLP	NTARAYT 5018
	J9 = 16	NTARAYT 5019
C		NTARAYT 5020
	A1 = SPAN*DLX (LIN) /2.	NTARAYT 5021
	A2 = A1	NTARAYT 5022
	A3 = SPAN*DLX (LCOOLP) /2.	NTARAYT 5023
	A4 = A3	NTARAYT 5024
	IF (IHC (IS-2) .EQ. 3) A1 = E*AREA (IS-2) /2.	NTARAYT 5025
	IF (IHC (IS-2) .EQ. 3) A3 = E*AREA (IS-1) /2.	NTARAYT 5026
	DX1 = DLX (L)	NTARAYT 5027
	DX2 = DLX (LDN)	NTARAYT 5028
	DX3 = DLX (LOUT)	NTARAYT 5029
	DX4 = DLX (LDN-2)	NTARAYT 5030
	DX5 = DLX (LIN)	NTARAYT 5031
	DX6 = DLX (LDN+1)	NTARAYT 5032
	DX7 = DLX (LCOOL)	NTARAYT 5033
	DX9 = DLX (LJ)	NTARAYT 5034
	DX10 = DLX (LDN-1)	NTARAYT 5035
C		NTARAYT 5036
	IF (IHC (ISUP) .EQ. 1) YIMPU = 1.0	NTARAYT 5037
	IF (IHC (ISUP) .EQ. 2) YCONVU = 1.0 + RCVRY*AM2 (ISUP) * (GAMC (ISUP) - 1.) /2.	NTARAYT 5038
	IF (IHC (ISUP) .EQ. 3) YFINSU = 1.0	NTARAYT 5039
C		NTARAYT 5040

IF (IS.LT.NSTA-1) GO TO 390	NTARAYT 5041
C	NTARAYT 5042
C FOR THE LAST STATIONS IN THE TRAILING EDGE:	NTARAYT 5043
C	NTARAYT 5044
DX2 = 0.0	NTARAYT 5045
DX4 = 0.0	NTARAYT 5046
DX6 = 0.0	NTARAYT 5047
DX10 = 0.0	NTARAYT 5048
390 CONTINUE	NTARAYT 5049
IF (IHC(IS).EQ.3) GO TO 420	NTARAYT 5050
IF (IHC(IS).EQ.2) GO TO 410	NTARAYT 5051
GO TO 340	NTARAYT 5052
C	NTARAYT 5053
C*** HCFPCD COMPUTES HC FOR FORCED CONVECTION	NTARAYT 5054
C	NTARAYT 5055
410 CONTINUE	NTARAYT 5056
TMP = (T(2,ISLICE,LCOOL) + T(2,ISLICE,LIN))/2.	NTARAYT 5057
CALL GASTBL(TMP,C,CP,GAM,PD,R,XMU)	NTARAYT 5058
RE(IS) = 12.*3600.*ABS(WCROS(2,ISLICE,IS))*DH(IS)/(A(LCOOL)*XMU)	NTARAYT 5059
C	NTARAYT 5060
HC(IS) = .023*12.*(C/DH(IS))*(RE(IS)**.8)*(PD**.333)	NTARAYT 5061
AHTRN1 = (DX5 + DX6)*SPAN/2.	NTARAYT 5062
GO TO 340	NTARAYT 5063
C	NTARAYT 5064
C***** SUBROUTINE HCPINS COMPUTES HC FOR A PIN FIN SURFACE OR FOR	NTARAYT 5065
C TURBULENT FORCED CONVECTION CHANNEL FLOW	NTARAYT 5066
420 CONTINUE	NTARAYT 5067
IF(ISENS.EQ.0) GO TO 424	NTARAYT 5068
AHTRN1 = EFAREA(IS)	NTARAYT 5069
HC(IS) = HC(IS-1)	NTARAYT 5070
GO TO 340	NTARAYT 5071
C	NTARAYT 5072
424 CALL HCPINS(IS,DELTAN,LCOOL,LCUP,LIN,LCOOLP,PINS,EFAREA)	NTARAYT 5073
AHTRN1 = EFAREA(IS)	NTARAYT 5074
IF (IS.GE.NSTA-1) AHTRN1 = AHTRN1/2.	NTARAYT 5075
A2 = EFAREA(IS)/2.	NTARAYT 5076
A4 = EFAREA(IS+1)/2.	NTARAYT 5077
GO TO 340	NTARAYT 5078
C	NTARAYT 5079
C	NTARAYT 5080
C	NTARAYT 5081
C	NTARAYT 5082
430 CONTINUE	NTARAYT 5083
440 CONTINUE	NTARAYT 5084
450 CONTINUE	NTARAYT 5085
C	NTARAYT 5086
RETURN	NTARAYT 5087
END	NTARAYT 5088
C----	
SOURCE.NTCOFTT	NTCOFTT 5089
SUBROUTINE TCOEF(IWRITE,WS,NIT,IPLLOT,ALPH2)	NTCOFTT 5090
C	NTCOFTT 5091
C- SOURCE.NTCOFTT	NTCOFTT 5092
C	NTCOFTT 5093
DIMENSION POLD(15,80), PSAV(5), X(80), ALPH2(4), DELTAN(15),	NTCOFTT 5094
Z TTOIC(80), JSO(15)	NTCOFTT 5095
DIMENSION PEXOLD(15), PIMOLD(15)	NTCOFTT 5096
REAL*8 TCOF	NTCOFTT 5097
COMMON /MATRX/ TCOF(400,30)	NTCOFTT 5098
C	NTCOFTT 5099
COMMON /PRPS/ CPO, GAMO, DP(80), SP(80), RE(80),	NTCOFTT 5100

```

C      Z      CPC (80),  GAMC (80),  DUMR1 (80),  DUMR2 (80)      NTCOFTT 5101
C      COMMON /TCO/ ADUMP,      BTA,      CD,      CP,      NTCOFTT 5102
C      Z      GAM,      PIM,      R,      SPAN,      TOG,      NTCOFTT 5103
C      Z      WDUMP,      WIM,      AKC (15,80),      AKW (15,80),      NTCOFTT 5104
C      Z      A (400),      AJET (80),      AM2 (80),      CNUM (80),      NTCOFTT 5105
C      Z      DH (80),      DHF (80),      DHJ (80),      NTCOFTT 5106
C      Z      DLX (400),      PF (80),      HC (80),      HG (80),      NTCOFTT 5107
C      Z      P (2,15,80),      PEXIT (15),      PUMP (80),      QG (80),      NTCOFTT 5108
C      Z      QSNK (80),      RR (80),      S (15),      T (2,15,400),      NTCOFTT 5109
C      Z      TG (80),      TAU (400),      WPC (80),      NTCOFTT 5110
C      Z      WJ (15,80),      WCROS (2,15,80),      XN (80),      NTCOFTT 5111
C      Z      ICOR,      IFILM,      IHUB,      ITIP,      NTCOFTT 5112
C      Z      ISBLOK,      ISLICE,      NBLKSZ,      NSLICE,      NTCOFTT 5113
C      Z      NFWD,      NSTA,      IHC (80),      NTCOFTT 5114
C      COMMON /TRNSNT/ RHOC,      RHOM,      SPHTC,      SPHTM,      NTCOFTT 5115
C      Z      DLTME,      TYME,      TEPS,      TYMMAX      NTCOFTT 5116
C      NTCOFTT 5117
C      NTCOFTT 5118
C      NTCOFTT 5119
C      NTCOFTT 5120
C      NTCOFTT 5121
C      NTCOFTT 5122
C      PROGRAM IS SET TO WORK WITH THE FOLLOWING UNITS ON THE VARIABLES:      NTCOFTT 5123
C      PRESSURE, P, IS IN PSIA *****      NTCOFTT 5124
C      FLOW RATES, WC,WJ,&WCROS ARE IN LBM/SEC *****      NTCOFTT 5125
C      AREAS ARE IN IN**2 *****      NTCOFTT 5126
C      ALL LENGTHS ARE IN INCHES *****      NTCOFTT 5127
C      HEAT TRANSFER COEFFICIENTS IN BTU/(HR*FT**2*R) *****      NTCOFTT 5128
C      NTCOFTT 5129
C      BTA = 0. INDICATES THAT A HEAT TRANSFER COEFFICIENT BOUNDARY      NTCOFTT 5130
C      CONDITION IS SPECIFIED      NTCOFTT 5131
C      1. INDICATES THAT A HEAT FLUX IS SPECIFIED ON THE GAS BOUNDARY      NTCOFTT 5132
C      UNITS EXPECTED ON THE FOLLOWING INPUT DATA ARE:      NTCOFTT 5133
C      LENGTHS ARE ALL IN INCHES-- DLX, TAU, SPAN, DH, DPX      NTCOFTT 5134
C      TEMPERATURES ARE ABSOLUTE (R)      NTCOFTT 5135
C      MASS FLOWS ARE ALL IN (LBM/SEC),--- WCROS, WJ, WPC, WDUMP      NTCOFTT 5136
C      HEAT TRANSFER COEFFICIENTS ARE IN BTU/(HR*FT**2*R),--HG,HC--GAS      NTCOFTT 5137
C      SIDE AND COOLANT SIDE .      NTCOFTT 5138
C      HEAT FLUX QG IS IN BTU/(HR*FT**2)      NTCOFTT 5139
C      HEAT SINK, QSNK, IS IN BTU/(HR)      NTCOFTT 5140
C      THERMAL PROPERTIES ARE: CONDUCTIVITY-AKW-BTU/(HR*FT*R)      NTCOFTT 5141
C      HEAT CAPACITY-CP-BTU/(LBM*R)      NTCOFTT 5142
C      NTCOFTT 5143
C      NTCOFTT 5144
C----SET UP FIRST GUESS AT TEMPERATURE DISTRIBUTION      NTCOFTT 5145
C----ASSUME COOLANT TEMPERATURE IS CONSTANT, = PLENUM STATIC TEMPERATURE      NTCOFTT 5146
C----ASSUME METAL TEMPERATURE IS CONSTANT, = 2200. R      NTCOFTT 5147
C      NTCOFTT 5148
C      NTCOFTT 5149
C      NTCOFTT 5150
C      NTCOFTT 5151
C      NTCOFTT 5152
C      NTCOFTT 5153
C      NTCOFTT 5154
C      NTCOFTT 5155
C      NTCOFTT 5156
C      NTCOFTT 5157
C      NTCOFTT 5158
C      NTCOFTT 5159
C      NTCOFTT 5160

```


C-- FOR TRANSIENT CASES, ADJUST INITIAL GUESS OF PRESSURE DISTRIBUTION	NTCOFTT 5161
C-- BASED ON THE VARIATION OF SUPPLY AND EXIT PRESSURES.	NTCOFTT 5162
C	NTCOFTT 5163
IF (TYME.GT.0.0) GO TO 234	NTCOFTT 5164
C	NTCOFTT 5165
C-- FOR STEADY STATE, ONLY INITIALIZE P'S ON FIRST OVERALL LOOP.	NTCOFTT 5166
C	NTCOFTT 5167
IF (NIT.GT.1) GO TO 255	NTCOFTT 5168
PEXOLD(ISLICE) = PEXIT(ISLICE)	NTCOFTT 5169
PIMOLD(ISLICE) = PIM	NTCOFTT 5170
160 WRITE(6,165) (I,T(1,ISLICE,I),I=1,NODST)	NTCOFTT 5171
165 FORMAT(1H1,' ASSUMED INITIAL TEMPERATURE DISTRIBUTION, (NODE NO.',	NTCOFTT 5172
Z ' ,T) '/7(' (' ,I3,',',P8.2,')',')	NTCOFTT 5173
170 CONTINUE	NTCOFTT 5174
C	NTCOFTT 5175
C----PRESSURE INITIALIZATION, GIVEN PIM (PLENUM STATIC PRESSURE) AND	NTCOFTT 5176
C----PEXIT (GAS SIDE STATIC PRESSURE AT TRAILING EDGE), FIT PRESSURE TO	NTCOFTT 5177
C A	NTCOFTT 5178
C----CUBIC EQN. OF THE FORM $A+B*X**3$, ASSUMING 85% OF THE PRESSURE DROP	NTCOFTT 5179
C----OCCURS IN THE TRAILING EDGE CHANNEL	NTCOFTT 5180
C	NTCOFTT 5181
X(1) = 0.0	NTCOFTT 5182
X(2) = DLX(10)	NTCOFTT 5183
DO 180 I = 3,NFWD	NTCOFTT 5184
X(I) = DLX(5*I) + X(I-2)	NTCOFTT 5185
180 CONTINUE	NTCOFTT 5186
P(1,ISLICE,1) = PIM - (PIM-PEXIT(ISLICE))*.15	NTCOFTT 5187
P(1,ISLICE,N) = PEXIT(ISLICE)	NTCOFTT 5188
P(1,ISLICE,N+1) = P(1,ISLICE,N)	NTCOFTT 5189
P(1,ISLICE,NFWD-1) = P(1,ISLICE,1) - .2*(P(1,ISLICE,1) - PEXIT(ISLICE))	NTCOFTT 5190
190 P(1,ISLICE,NFWD) = P(1,ISLICE,NFWD-1)	NTCOFTT 5191
DO 200 I = 3,NFWD,2	NTCOFTT 5192
P(1,ISLICE,I) = P(1,ISLICE,1) - (P(1,ISLICE,1) - P(1,ISLICE,NFWD))*	NTCOFTT 5193
Z (X(I)/X(NFWD))**3	NTCOFTT 5194
200 P(1,ISLICE,I-1) = P(1,ISLICE,1) - (P(1,ISLICE,1) - P(1,ISLICE,NFWD-1))*	NTCOFTT 5195
Z (X(I-1)/X(NFWD-1))**3	NTCOFTT 5196
ISTR = NFWD+1	NTCOFTT 5197
IFNL = N	NTCOFTT 5198
C	NTCOFTT 5199
C----FOR TRAILING EDGE CHANNEL, X VALUES ARE RELATIVE TO END OF	NTCOFTT 5200
C IMPINGEMENT CHANNEL	NTCOFTT 5201
C	NTCOFTT 5202
X(NFWD+1) = (DLX(NODSF) + DLX(NODSF-5))/2.	NTCOFTT 5203
210 ITEM = NFWD+3	NTCOFTT 5204
DO 220 I = ITEM,N,2	NTCOFTT 5205
LCOOL = 5*I	NTCOFTT 5206
X(I) = X(I-2) + DLX(LCOOL)	NTCOFTT 5207
220 CONTINUE	NTCOFTT 5208
DO 230 I = ISTR,IFNL,2	NTCOFTT 5209
P(1,ISLICE,I) = P(1,ISLICE,NFWD) - (P(1,ISLICE,NFWD) - PEXIT(ISLICE))*	NTCOFTT 5210
Z (X(I)/X(N))**3	NTCOFTT 5211
230 P(1,ISLICE,I+1) = P(1,ISLICE,I)	NTCOFTT 5212
DO 232 I = 1,NSTA	NTCOFTT 5213
232 POLD(ISLICE,I) = P(1,ISLICE,I)	NTCOFTT 5214
DO 233 I = 1,NSTA	NTCOFTT 5215
233 P(2,ISLICE,I) = P(1,ISLICE,I)	NTCOFTT 5216
GO TO 240	NTCOFTT 5217
C	NTCOFTT 5218
234 DLTAPC = .84*(PIM-PIMOLD(ISLICE))	NTCOFTT 5219
PIMOLD(ISLICE) = PIM	NTCOFTT 5220

	DLTAPE = PEXIT(ISLICE) - PEXOLD(ISLICE)	NTCOFTT 5221
	PEXOLD(ISLICE) = PEXIT(ISLICE)	NTCOFTT 5222
	DO 235 I = 1, NPWD	NTCOFTT 5223
235	P(2, ISLICE, I) = P(2, ISLICE, I) + DLTAPE	NTCOFTT 5224
	ISTRT = NPWD + 1	NTCOFTT 5225
	IFNL = NSTA - 1	NTCOFTT 5226
	DO 236 I = ISTRT, IFNL, 2	NTCOFTT 5227
	P(2, ISLICE, I) = P(2, ISLICE, I) + DLTAPE * (1.0 - X(I) / X(IFNL)) +	NTCOFTT 5228
	Z DLTAPE * X(I) / X(IFNL)	NTCOFTT 5229
236	P(2, ISLICE, I + 1) = P(2, ISLICE, I)	NTCOFTT 5230
	DO 237 I = 1, NSTA	NTCOFTT 5231
237	POLD(ISLICE, I) = P(2, ISLICE, I)	NTCOFTT 5232
	GO TO 255	NTCOFTT 5233
C		NTCOFTT 5234
240	WRITE(6, 245)	NTCOFTT 5235
245	FORMAT(/' INITIAL PRESSURE DIST. (STATION NO., P) ' /)	NTCOFTT 5236
	WRITE(6, 250) (I, P(1, ISLICE, I), I = 1, ISTRT)	NTCOFTT 5237
	WRITE(6, 250) (I, P(1, ISLICE, I), I = ITEM, N, 2)	NTCOFTT 5238
250	FORMAT(7(' (', I3, ',', ' ', P7.2, ')', ' '))	NTCOFTT 5239
C		NTCOFTT 5240
C		NTCOFTT 5241
C		NTCOFTT 5242
255	CONTINUE	NTCOFTT 5243
	DO 260 I = 1, 4	NTCOFTT 5244
260	PSAV(I) = 0.0	NTCOFTT 5245
C		NTCOFTT 5246
C----	INITIALLY, THE FLOW SPLIT AT THE LEADING EDGE IS ASSUMED	NTCOFTT 5247
C	TO BE 50/50 (DELTA = .5)	NTCOFTT 5248
C		NTCOFTT 5249
C----	IDELT COUNTS THE NUMBER OF FLOW SPLIT ITERATIONS. IF NO	NTCOFTT 5250
C	CONVERGENCE, IDELT IS SET NEGATIVE.	NTCOFTT 5251
C----	DELTAN IS THE FRACTION OF FLOW TO THE SUCTION SIDE (EVEN	NTCOFTT 5252
C	NUMBERED STATIONS)	NTCOFTT 5253
C----	IVERGE COUNTS THE NUMBER OF ITERATIONS AT A GIVEN FLOW SPLIT	NTCOFTT 5254
C		NTCOFTT 5255
C	IFNL = THE NUMBER OF FLOW CHANNEL NODES, USED IN PRESSURE CALCULATIONS	NTCOFTT 5256
C		NTCOFTT 5257
	IF (NIT.EQ.1.AND.TYME.LT.0.0) DELTAN(ISLICE) = .5	NTCOFTT 5258
275	CONTINUE	NTCOFTT 5259
	IFNL = NSTA - 3	NTCOFTT 5260
	IVERGE = 0	NTCOFTT 5261
	IDELT = 1	NTCOFTT 5262
	JS = 1	NTCOFTT 5263
	IF (NIT.GT.1) JS = JSO(ISLICE)	NTCOFTT 5264
290	CONTINUE	NTCOFTT 5265
	IVERGE = IVERGE + 1	NTCOFTT 5266
300	CONTINUE	NTCOFTT 5267
C		NTCOFTT 5268
	JSSENS = JS - 2 * (JS / 2)	NTCOFTT 5269
C		NTCOFTT 5270
C----	SUBROUTINE FLOWS COMPUTES JET FLOW RATES, CROSSFLOW RATES, AND	NTCOFTT 5271
C	THE SQUARE OF THE MACH NUMBER	NTCOFTT 5272
C		NTCOFTT 5273
310	CONTINUE	NTCOFTT 5274
C		NTCOFTT 5275
	CALL FLOWS(JS, DELTAN, ICHOK, AMCHOK)	NTCOFTT 5276
320	CONTINUE	NTCOFTT 5277
	IF (WJ(ISLICE, JS).LE.0.0) GO TO 370	NTCOFTT 5278
C		NTCOFTT 5279
C	SUBROUTINE HCOOL COMPUTES IMPINGEMENT REGION HEAT TRANSFER COEFFICIENTS	NTCOFTT 5280

C	CALL HCOOL(JS)	NTCOFTT 5281
330	CONTINUE	NTCOFTT 5282
	CALL THRCON	NTCOFTT 5283
C	SUBROUTINE THRCON EXTRACTS THERMAL CONDUCTIVITIES FROM INPUT	NTCOFTT 5284
C	TABLES AKCTBL AND AKWTBL.	NTCOFTT 5285
C		NTCOFTT 5286
335	CONTINUE	NTCOFTT 5287
C		NTCOFTT 5288
C---	CHECK TO MAKE SURE STAGNATION HC IS LESS THAN MAXIMUM PHYSICALLY	NTCOFTT 5289
C	POSSIBLE VALUE.	NTCOFTT 5290
C		NTCOFTT 5291
	IF (JS.EQ.1) ASTG = (DLX(9)+DLX(14))*SPAN/2.0	NTCOFTT 5292
	IF (JS.GT.1) ASTG = (DLX(5*JS-1)+DLX(5*JS+9))*SPAN/2.0	NTCOFTT 5293
	HSTGMX = WJ(ISLICE,JS)*CFO*144.*3600./ASTG	NTCOFTT 5294
	IF (HC(JS).GT.HSTGMX) HC(JS) = HSTGMX	NTCOFTT 5295
C		NTCOFTT 5296
C		NTCOFTT 5297
337	CONTINUE	NTCOFTT 5298
	CALL TAPRAY(JS,JSSENS,DELTAN)	NTCOFTT 5299
C		NTCOFTT 5300
C	SOLVE THE TCOF ARRAY AND SET NEW TEMPERATURE VALUES:	NTCOFTT 5301
C		NTCOFTT 5302
340	CONTINUE	NTCOFTT 5303
	CALL GAUSS(NODST,23)	NTCOFTT 5304
	DO 350 I = 1,NODST	NTCOFTT 5305
	T(2,ISLICE,I) = TCOF(I,24)	NTCOFTT 5306
	IF (T(2,ISLICE,I).LE.0.0) T(2,ISLICE,I) = T0G	NTCOFTT 5307
350	CONTINUE	NTCOFTT 5308
C*****		NTCOFTT 5309
360	IF (ABS((T(2,ISLICE,NSAVE)-TSAVE)/TSAVE).GT..05)	NTCOFTT 5310
Z	CALL FLOWS(JS,DELTAN,ICHOKE,AMCHOK)	NTCOFTT 5311
	TSAVE = T(2,ISLICE,NSAVE)	NTCOFTT 5312
C*****		NTCOFTT 5313
	IF (ICHOKE.EQ.0) GO TO 370	NTCOFTT 5314
	WRITE(8,365) ISLICE,IVERGE,IDELT,NIT,ICHOKE,AMCHOK	NTCOFTT 5315
365	FORMAT(/10X,'SLICE ',I2,' IS CHOKED FOR IVERGE =',I3,', IDELT =',	NTCOFTT 5316
Z	I3,', NIT =',I3,', ICHOKE =',I4,', M**2 =',F6.3)	NTCOFTT 5317
370	CONTINUE	NTCOFTT 5318
C		NTCOFTT 5319
C	COMPUTE NEW PRESSURES	NTCOFTT 5320
C		NTCOFTT 5321
	CALL PARRAY(JS,JSSENS,ICHOKE)	NTCOFTT 5322
C		NTCOFTT 5323
C	SOLVE THE TCOF ARRAY AND COMPUTE NEW PRESSURES	NTCOFTT 5324
C		NTCOFTT 5325
430	CONTINUE	NTCOFTT 5326
	CALL GAUSS(IFNL,19)	NTCOFTT 5327
440	CONTINUE	NTCOFTT 5328
	DO 460 I = 1,IFNL	NTCOFTT 5329
450	P(2,ISLICE,I) = TCOF(I,20)	NTCOFTT 5330
460	CONTINUE	NTCOFTT 5331
	P(2,ISLICE,IFNL+1) = P(2,ISLICE,IFNL)	NTCOFTT 5332
	P(2,ISLICE,NSTA-1) = PEXIT(ISLICE)	NTCOFTT 5333
	P(2,ISLICE,NSTA) = PEXIT(ISLICE)	NTCOFTT 5334
470	CONTINUE	NTCOFTT 5335
	IF (IWRITE.EQ.2) CALL WR0UT(IDELT,JS,DELTAN,IVERGE)	NTCOFTT 5336
	IF (IPL0T.EQ.2) CALL PLOTMF(ALPH2)	NTCOFTT 5337
C		NTCOFTT 5338
480	CONTINUE	NTCOFTT 5339
		NTCOFTT 5340

C		NTCOFTT	5341
C	CHECK OVERALL CONVERGENCE	NTCOFTT	5342
C	CALCULATIONS ARE REPEATED UNTIL THE PRESSURE AT STATION 1 (NODE 5)	NTCOFTT	5343
C	HAS STABILIZED FOR	NTCOFTT	5344
C	FOUR ITERATIONS. THEN WE GO TO CHECK THE FLOW SPLIT.	NTCOFTT	5345
C		NTCOFTT	5346
	DO 490 I=1,3	NTCOFTT	5347
	K=5-I	NTCOFTT	5348
490	PSAV(K)=PSAV(K-1)	NTCOFTT	5349
	PSAV(1)=P(2,ISLICE,JS)	NTCOFTT	5350
C		NTCOFTT	5351
	DIFO=0.0	NTCOFTT	5352
	DO 500 I=1,3	NTCOFTT	5353
	JJ=I+1	NTCOFTT	5354
	DO 500 K=JJ,4	NTCOFTT	5355
	DIFN=ABS(PSAV(I)-PSAV(K))	NTCOFTT	5356
	IF(DIFO.LT.DIFN)DIFO=DIFN	NTCOFTT	5357
500	CONTINUE	NTCOFTT	5358
C		NTCOFTT	5359
510	DIFO=DIFO/(PIM-PEXIT(ISLICE))	NTCOFTT	5360
	PCNVRG = .01	NTCOFTT	5361
	IF (NIT.EQ.1.AND.NSLICE.GT.1) PCNVRG = .05	NTCOFTT	5362
	EPSN = (P(2,ISLICE,NFWD-1)-P(2,ISLICE,NFWD))/(P(2,ISLICE,NFWD-1))	NTCOFTT	5363
	IF (IDELT.EQ.1.AND.IVERGE.LT.8.AND.TYME.LE.0.0) GO TO 516	NTCOFTT	5364
	IF (DIFO.LE.PCNVRG.AND.IVERGE.GE.4) GO TO 560	NTCOFTT	5365
516	CONTINUE	NTCOFTT	5366
	IF (IVERGE.LT.10) GO TO 520	NTCOFTT	5367
	IF (ABS(PSAV(1)-PSAV(2)).GT.ABS(PSAV(3)-PSAV(4))) V=1.-(1.-V)/2.	NTCOFTT	5368
520	CONTINUE	NTCOFTT	5369
	DO 530 I = 1,NSTA	NTCOFTT	5370
	P(2,ISLICE,I) = P(2,ISLICE,I) + V*(POLD(ISLICE,I) - P(2,ISLICE,I))	NTCOFTT	5371
	IF (P(2,ISLICE,I).GE.PIM) P(2,ISLICE,I) = .999*PIM	NTCOFTT	5372
	POLD(ISLICE,I) = P(2,ISLICE,I)	NTCOFTT	5373
	TTOTC(I) = T(2,ISLICE,5*I)*(1.+(GAMC(I)-1.)*AM2(I)/2.)	NTCOFTT	5374
530	CONTINUE	NTCOFTT	5375
C		NTCOFTT	5376
540	CONTINUE	NTCOFTT	5377
	IF (IVERGE.GT.30.OR.V.G1..95)	NTCOFTT	5378
Z	WRITE(8,550) IVERGE,V,(PSAV(I),I=1,4)	NTCOFTT	5379
550	FORMAT(' ***** CONVERGENCE PROBLEMS *****'/	NTCOFTT	5380
Z	' IVERGE=',I3,'; V= ',F6.4,'; PSAV= ',4(F10.2))	NTCOFTT	5381
	IF (IVERGE.GT.50) GO TO 590	NTCOFTT	5382
	GO TO 290	NTCOFTT	5383
C		NTCOFTT	5384
C	C***** ONCE PRESSURE-TEMPERATURE ITERATION HAS CONVERGED, CHECK	NTCOFTT	5385
C	THE FLOW SPLIT AND ADJUST IF NECESSARY.	NTCOFTT	5386
C		NTCOFTT	5387
560	CONTINUE	NTCOFTT	5388
	IF (IWRITE.EQ.1) CALL WROUT(IDELT,JS,DELTAN,IVERGE)	NTCOFTT	5389
	IF (ICHOKE.GT.0) WRITE(6,565) ICHOKE,AMCHOK	NTCOFTT	5390
565	FORMAT(/10X,'MACH NO. > 1 AT STATION ',I4,', M**2 =',F6.3/)	NTCOFTT	5391
570	CONTINUE	NTCOFTT	5392
	EPSN = (P(2,ISLICE,NFWD-1)-P(2,ISLICE,NFWD))/(P(2,ISLICE,NFWD-1))	NTCOFTT	5393
575	CONTINUE	NTCOFTT	5394
	IF (TYME.GT.0.0) GO TO 590	NTCOFTT	5395
	CALL FLSPLT(AJET,EPSN,ISLICE,NODSF,IDELT,JS,DELTAN,ICONV)	NTCOFTT	5396
580	CONTINUE	NTCOFTT	5397
	IF (ICONV.EQ.1) CALL WROUT(IDELT,JS,DELTAN,IVERGE)	NTCOFTT	5398
	IVERGE = 0	NTCOFTT	5399
	IF (ICONV.EQ.0) GO TO 290	NTCOFTT	5400


```

590  CONTINUE
      IF (TYME.GT.0.0) CALL WROUT(IDELT,JS,DELTAN,IVERGE)
      JSO(ISLICE) = JS
      RETURN
      END
NTCOFTT 5401
NTCOFTT 5402
NTCOFTT 5403
NTCOFTT 5404
NTCOFTT 5405

C----SOURCE.NTHRCNT
      SUBROUTINE THRCON
NTHRCNT 5406
NTHRCNT 5407
NTHRCNT 5408
C- SOURCE.NTHRCNT----
NTHRCNT 5409
NTHRCNT 5410
C
      COMMON /BOUND/ BCXS(400), BCXP(400), BCHGS(1000), BCHGF(1000),
Z      BCTGS(1000), BCTGP(1000), BCQGS(1000), BCQGP(1000),
Z      BCPGS(1000), BCPGP(1000), THUBIN(400), THUB(80),
Z      QHUBIN(400), OHUB(80), TTIPIN(400), TTIP(80),
Z      QTIPIN(400), QTIP(80), RHOVG(400), PEX(400),
Z      BCTIME(50), TTIO(50), PTIO(50), WPLEN,
Z      WSVST(50), AKCTBL(20), AKWTBL(20), NBCS, NBCP
NTHRCNT 5411
NTHRCNT 5412
NTHRCNT 5413
NTHRCNT 5414
NTHRCNT 5415
NTHRCNT 5416
NTHRCNT 5417
NTHRCNT 5418
C
      COMMON /TCO/ ADUMP,      BTA,      CD,      CP,
Z      GAM,      FIM,      R,      SPAN,      TOG,
Z      WDUMP,      WIM,      AKC(15,80), AKW(15,80),
Z      A(400),      AJET(80),      AM2(80),      CNUM(80),
Z      DH(80),      DHF(80),      DHJ(80),
Z      DLX(400),      PF(80),      HC(80),      HG(80),
Z      P(2,15,80), PEXIT(15),      PUMP(80),      QG(80),
Z      QSNK(80),      PR(80),      S(15),      T(2,15,400),
Z      TG(80),      TAU(400),      WFC(80),
Z      WJ(15,80),      WCROS(2,15,80),      XN(80),
Z      ICOF,      IFILM,      IHUB,      ITIP,
Z      ISBLOK,      ISLICE,      NBLKSZ,      NSLICE,
Z      NFWD,      NSTA,      IHC(80)
NTHRCNT 5419
NTHRCNT 5420
NTHRCNT 5421
NTHRCNT 5422
NTHRCNT 5423
NTHRCNT 5424
NTHRCNT 5425
NTHRCNT 5426
NTHRCNT 5427
NTHRCNT 5428
NTHRCNT 5429
NTHRCNT 5430
NTHRCNT 5431
NTHRCNT 5432
NTHRCNT 5433
NTHRCNT 5434
NTHRCNT 5435
NTHRCNT 5436
NTHRCNT 5437
NTHRCNT 5438
NTHRCNT 5439
NTHRCNT 5440
NTHRCNT 5441
NTHRCNT 5442
NTHRCNT 5443
NTHRCNT 5444
NTHRCNT 5445
NTHRCNT 5446
NTHRCNT 5447
NTHRCNT 5448
NTHRCNT 5449
NTHRCNT 5450
NTHRCNT 5451
NTHRCNT 5452
NTHRCNT 5453
NTHRCNT 5454
NTHRCNT 5455
NTHRCNT 5456
NTHRCNT 5457
NTHRCNT 5458
NTHRCNT 5459
NTHRCNT 5460

      DO 1000 I = 1,NSTA
      L = 5*I - 2
      LJ = L - 1
      LOUT = L - 2
      TC = (T(2,ISLICE,LJ) + T(2,ISLICE,LOUT))/2.0 - 460.
      TW = T(2,ISLICE,L) - 460.

C
C LOOK UP COATING THERMAL CONDUCTIVITY IN TABLE AKCTBL.
C
      IF (TC.GT.AKCTBL(1)) GO TO 150
C
C FOR A TEMPERATURE LOWER THAN THE BOTTOM OF THE TABLE, EXTRAPOLATE
C BELOW TABLE
      RATIO = (TC - AKCTBL(1))/(AKCTBL(3) - AKCTBL(1))
      AKC(ISLICE,1) = AKCTBL(2) + (AKCTBL(4) - AKCTBL(2))*RATIO
      GO TO 500
C
150  CONTINUE
C
C FIND SIZE OF TABLE
C
      DO 152 J = 3,19,2
      JLST = J-1
      IF (AKCTBL(J).LE.0.1) GO TO 154
152  CONTINUE
154  JLSTM = JLST-1
C
C LOCATE WHERE TEMPERATURE FALLS IN THE TABLE AKCTBL.

```

C	DO 170 J = 3, JLSTH, 2	NTHRCNT 5461
	IF (TC.GT.AKCTBL(J)) GO TO 160	NTHRCNT 5462
C		NTHRCNT 5463
C	FOUND LOCATION, NOW INTERPOLATE.	NTHRCNT 5464
C		NTHRCNT 5465
	RATIO = (TC - AKCTBL(J-2))/(AKCTBL(J) - AKCTBL(J-2))	NTHRCNT 5466
	AKC(ISLICE, I) = AKCTBL(J-1) + (AKCTBL(J+1) - AKCTBL(J-1))*RATIO	NTHRCNT 5467
	GO TO 500	NTHRCNT 5468
160	IF (J.LT.JLSTH) GO TO 170	NTHRCNT 5469
C		NTHRCNT 5470
C	TEMPERATURE IS ABOVE THE RANGE OF THE TABLE, SO EXTRAPOLATE UP.	NTHRCNT 5471
C		NTHRCNT 5472
	RATIO = (TC - AKCTBL(J-2))/(AKCTBL(J) - AKCTBL(J-2))	NTHRCNT 5473
	AKC(ISLICE, I) = AKCTBL(J-1) + (AKCTBL(J+1) - AKCTBL(J-1))*RATIO	NTHRCNT 5474
	GO TO 500	NTHRCNT 5475
170	CONTINUE	NTHRCNT 5476
500	CONTINUE	NTHRCNT 5477
C		NTHRCNT 5478
C	NOW LOOK UP METAL CONDUCTIVITY IN TABLE AKWTBL.	NTHRCNT 5479
C		NTHRCNT 5480
	IF (TW.GT.AKWTBL(1)) GO TO 550	NTHRCNT 5481
C		NTHRCNT 5482
C	FOR A TEMPERATURE LOWER THAN THE BOTTOM OF THE TABLE, EXTRAPOLATE	NTHRCNT 5483
C	BELOW TABLE	NTHRCNT 5484
	RATIO = (TW - AKWTBL(1))/(AKWTBL(3) - AKWTBL(1))	NTHRCNT 5485
	AKW(ISLICE, I) = AKWTBL(2) + (AKWTBL(4) - AKWTBL(2))*RATIO	NTHRCNT 5486
	GO TO 1000	NTHRCNT 5487
C		NTHRCNT 5488
550	CONTINUE	NTHRCNT 5489
C		NTHRCNT 5490
C	FIND SIZE OF TABLE	NTHRCNT 5491
C		NTHRCNT 5492
	DO 552 J = 3, 19, 2	NTHRCNT 5493
	JLST = J-1	NTHRCNT 5494
	IF (AKWTBL(J).LE.0.1) GO TO 554	NTHRCNT 5495
552	CONTINUE	NTHRCNT 5496
554	JLSTH = JLST-1	NTHRCNT 5497
C		NTHRCNT 5498
C	LOCATE WHERE TEMPERATURE FALLS IN THE TABLE AKWTBL.	NTHRCNT 5499
C		NTHRCNT 5500
	DO 570 J = 3, JLSTH, 2	NTHRCNT 5501
	IF (TW.GT.AKWTBL(J)) GO TO 560	NTHRCNT 5502
C		NTHRCNT 5503
C	FOUND LOCATION, NOW INTERPOLATE.	NTHRCNT 5504
C		NTHRCNT 5505
	RATIO = (TW - AKWTBL(J-2))/(AKWTBL(J) - AKWTBL(J-2))	NTHRCNT 5506
	AKW(ISLICE, I) = AKWTBL(J-1) + (AKWTBL(J+1) - AKWTBL(J-1))*RATIO	NTHRCNT 5507
	GO TO 1000	NTHRCNT 5508
560	IF (J.LT.JLSTH) GO TO 570	NTHRCNT 5509
C		NTHRCNT 5510
C	TEMPERATURE IS ABOVE THE RANGE OF THE TABLE, SO EXTRAPOLATE UP.	NTHRCNT 5511
C		NTHRCNT 5512
	RATIO = (TW - AKWTBL(J-2))/(AKWTBL(J) - AKWTBL(J-2))	NTHRCNT 5513
	AKW(ISLICE, I) = AKWTBL(J-1) + (AKWTBL(J+1) - AKWTBL(J-1))*RATIO	NTHRCNT 5514
	GO TO 1000	NTHRCNT 5515
570	CONTINUE	NTHRCNT 5516
1000	CONTINUE	NTHRCNT 5517
	RETURN	NTHRCNT 5518
	END	NTHRCNT 5519
		NTHRCNT 5520

```

C-----SOURCE.NWROTTT
      SUBROUTINE WROUT (IDELT,JS,DELTAN,IVERGE)
C
C- SOURCE.NWROTTT-----
C
C DUMR2 CARRIES THE IMPINGEMENT JET REYNOLDS NO. IN FROM HCCOOL.
C
      COMMON /CHKHOL/ WCHK(80), WCHKDM
C
      COMMON /FLMCOL/ RHOVGA(80), FG(80), XFC(80), FLMEFF(80),
Z      XMUC(80), FMES(80), REFC(80), NFCSUP(80)
C
      COMMON /PPPS/ CPO, GAMO, DP(80), SP(80), RE(80),
Z      CPC(80), GAMC(80), DUMR1(80), DUMR2(80)
C
      COMMON /EADL/ APLN(15), DPLN(15), FIN(15), ROUT(15),
Z      PIN(15), TIN(15), W(15), WS
C
      COMMON /TCO/ ADUMP, BTA, CD, CP,
Z      GAM, FIM, R, SPAN, TOG,
Z      WDUMP, WIM, AKC(15,80), AKW(15,80),
Z      A(400), AJET(80), AM2(80), CNUM(80),
Z      DH(80), DHF(80), DHJ(80),
Z      DLX(400), FF(80), HC(80), HG(80),
Z      P(2,15,80), PEXIT(15), PUMP(80), QG(80),
Z      QSNK(80), RR(80), S(15), T(2,15,400),
Z      TG(80), TAU(400), WFC(80),
Z      WJ(15,80), WCROS(2,15,80), XN(80),
Z      ICOR, IFILM, IHUB, ITIP,
Z      ISBLOK, ISLICE, NBLKSZ, NSLICE,
Z      NPWD, NSTA, IHC(80)
C
      COMMON /TRNSNT/ RHOC, RHOM, SPHTC, SPHTM,
Z      DLTYME, TYME, TEPS, TYMMAX
C
      COMMON /UNITS/ CINCH(2), CHTC(2), CHFLX(2), CPRSR(2), CMSFL(2),
Z      CTMPF(2), CTCGN(2), CDEN(2), CSPHT(2), CGASC(2),
Z      CVISC(2), CRHOVG(2), IUNITS
C
      DIMENSION DUM1(10),DUM2(10),DELTAN(15)
10 CONTINUE
      IF (ISLICE.EQ.1) TBULK = 0.0
      IF (ISLICE.EQ.1) TOTSPN = 0.0
      TTYME = TYME
      IF (TTYME.LT.0.) TTYME=0.0
      WRITE(6,90) TTYME,DLTYME,WS
90 FORMAT(1H1,10X,' TIME = ',F6.2,' SEC., STEP SIZE = ',F6.3,
Z      ' SEC., WHEEL SPEED = ',F8.1,' RPM')
8805 WRITE(6,8806) ISLICE,IDELT,JS,DELTAN(ISLICE),IVERGE
      ITPBG = NPWD + 2
      WRITE(6,100) ITRBG
C
      IF (IUNITS.EQ.2) WRITE(6,270)
      IF (IUNITS.EQ.1) WRITE(6,271)
      DO 210 I = 1,NSTA,2
      II = I
      LCOOL = 5*II
      NOS = LCOOL - 4
      DO 205 J = 1,4
      JM = NOS+J-1

```

```

NWROTTT 5521
NWROTTT 5522
NWROTTT 5523
NWROTTT 5524
NWROTTT 5525
NWROTTT 5526
NWROTTT 5527
NWROTTT 5528
NWROTTT 5529
NWROTTT 5530
NWROTTT 5531
NWROTTT 5532
NWROTTT 5533
NWROTTT 5534
NWROTTT 5535
NWROTTT 5536
NWROTTT 5537
NWROTTT 5538
NWROTTT 5539
NWROTTT 5540
NWROTTT 5541
NWROTTT 5542
NWROTTT 5543
NWROTTT 5544
NWROTTT 5545
NWROTTT 5546
NWROTTT 5547
NWROTTT 5548
NWROTTT 5549
NWROTTT 5550
NWROTTT 5551
NWROTTT 5552
NWROTTT 5553
NWROTTT 5554
NWROTTT 5555
NWROTTT 5556
NWROTTT 5557
NWROTTT 5558
NWROTTT 5559
NWROTTT 5560
NWROTTT 5561
NWROTTT 5562
NWROTTT 5563
NWROTTT 5564
NWROTTT 5565
NWROTTT 5566
NWROTTT 5567
NWROTTT 5568
NWROTTT 5569
NWROTTT 5570
NWROTTT 5571
NWROTTT 5572
NWROTTT 5573
NWROTTT 5574
NWROTTT 5575
NWROTTT 5576
NWROTTT 5577
NWROTTT 5578
NWROTTT 5579
NWROTTT 5580

```

	DUM1(J) = T(2,ISLICE,JM) - 460.	NWROTTT 5581
	IF (IUNITS.EQ.1) DUM1(J) = T(2,ISLICE,JM)/CTMPF(1)	NWROTTT 5582
205	CONTINUE	NWROTTT 5583
	DUM1(5) = T(2,ISLICE,LCOOL) - 460.	NWROTTT 5584
	IF (IUNITS.EQ.1) DUM1(5) = T(2,ISLICE,LCOOL)/CTMPF(1)	NWROTTT 5585
	DUM1(6) = P(2,ISLICE,II)/CPRSF(IUNITS)	NWROTTT 5586
	DUM1(7) = DUM1(6)*(1.+(GAMC(II)-1.)*AM2(II)/2.)*(GAMC(II)/(GAMC(II)-1.))	NWROTTT 5587
	Z	NWROTTT 5588
	DUM1(8) = HC(II)/CHTC(IUNITS)	NWROTTT 5589
	DUM1(9) = HG(II)/CHTC(IUNITS)	NWROTTT 5590
	DUM1(10) = TG(II) - 460.	NWROTTT 5591
	IF (IUNITS.EQ.1) DUM1(10) = TG(II)/1.8	NWROTTT 5592
	IF (BTA.GT..001) DUM1(10) = 9.E20	NWROTTT 5593
	IF (BTA.GT..001) DUM1(9) = 9.E20	NWROTTT 5594
	WRITE(6,274) (II,LCOOL,(DUM1(J),J=1,10))	NWROTTT 5595
	IF (I.EQ.NFWD) WRITE(6,276)	NWROTTT 5596
210	CONTINUE	NWROTTT 5597
	IF (IUNITS.EQ.2) WRITE(6,278)	NWROTTT 5598
	IF (IUNITS.EQ.1) WRITE(6,279)	NWROTTT 5599
	DO 220 I = 1,NSTA,2	NWROTTT 5600
	II = I	NWROTTT 5601
	LCOOL = 5*II	NWROTTT 5602
	NOS = LCOOL - 4	NWROTTT 5603
	DUM2(1) = WJ(ISLICE,II)/CMSFL(IUNITS)	NWROTTT 5604
	DUM2(2) = DUMR2(II)	NWROTTT 5605
	DUM2(3) = WCROS(2,ISLICE,II)/CMSFL(IUNITS)	NWROTTT 5606
	DUM2(4) = RE(II)	NWROTTT 5607
	DUM2(5) = SQRT(AM2(II))	NWROTTT 5608
	DUM2(6) = FF(II)	NWROTTT 5609
	DUM2(7) = WFC(II)/CMSFL(IUNITS)	NWROTTT 5610
	DUM2(8) = FLMEFF(II)	NWROTTT 5611
	WRITE(6,280) (II,LCOOL,DUM2(1),WCHK(II),(DUM2(J),J=2,8))	NWROTTT 5612
	IF (I.EQ.NFWD) WRITE(6,276)	NWROTTT 5613
220	CONTINUE	NWROTTT 5614
	DUM2(9) = WDUMP/CMSFL(IUNITS)	NWROTTT 5615
	IF (ADUMP.GT.0.0.AND.IUNITS.EQ.2) WRITE(6,290) DUM2(9),WCHKDM	NWROTTT 5616
	IF (ADUMP.GT.0.0.AND.IUNITS.EQ.1) WRITE(6,291) DUM2(9),WCHKDM	NWROTTT 5617
C		NWROTTT 5618
	ITRBG = NFWD + 1	NWROTTT 5619
	WRITE(6,124) ISLICE,ITRBG	NWROTTT 5620
C		NWROTTT 5621
	IF (IUNITS.EQ.2) WRITE(6,270)	NWROTTT 5622
	IF (IUNITS.EQ.1) WRITE(6,271)	NWROTTT 5623
	DO 230 I = 1,NSTA,2	NWROTTT 5624
	II = I	NWROTTT 5625
	IF (I.GT.1) II = I-1	NWROTTT 5626
	LCOOL = 5*II	NWROTTT 5627
	NOS = LCOOL - 4	NWROTTT 5628
	DO 225 J = 1,5	NWROTTT 5629
	JM = NOS+J-1	NWROTTT 5630
	DUM1(J) = T(2,ISLICE,JM) - 460.	NWROTTT 5631
	IF (IUNITS.EQ.1) DUM1(J) = T(2,ISLICE,JM)/CTMPF(1)	NWROTTT 5632
225	CONTINUE	NWROTTT 5633
	DUM1(6) = P(2,ISLICE,II)/CPRSF(IUNITS)	NWROTTT 5634
	DUM1(7) = DUM1(6)*(1.+(GAMC(II)-1.)*AM2(II)/2.)*(GAMC(II)/(GAMC(II)-1.))	NWROTTT 5635
	Z	NWROTTT 5636
	DUM1(8) = HC(II)/CHTC(IUNITS)	NWROTTT 5637
	DUM1(9) = HG(II)/CHTC(IUNITS)	NWROTTT 5638
	DUM1(10) = TG(II) - 460.	NWROTTT 5639
	IF (IUNITS.EQ.1) DUM1(10) = TG(II)/1.8	NWROTTT 5640


```

IF (BTA.GT..001) DUM1(10) = 9.E20
IF (BTA.GT..001) DUM1(9) = 9.E20
WRITE(6,274) (II,LCOOL,(DUM1(J),J=1,10))
IF (I.EQ.NFWD) WRITE(6,276)
230 CONTINUE
IF (IUNITS.EQ.2) WRITE(6,278)
IF (IUNITS.EQ.1) WRITE(6,279)
DO 240 I = 1,NSTA,2
II = I
IF (I.GT.1) II = I-1
LCOOL = 5*II
NOS = LCOOL - 4
DUM2(1) = WJ(ISLICE,II)/CMSFL(IUNITS)
DUM2(2) = DUMR2(II)
DUM2(3) = WCROS(2,ISLICE,II)/CMSFL(IUNITS)
DUM2(4) = RE(II)
DUM2(5) = SQRT(AM2(II))
DUM2(6) = FF(II)
DUM2(7) = WPC(II)/CMSFL(IUNITS)
DUM2(8) = FLMEFF(II)
WRITE(6,280) (II,LCOOL,DUM2(1),WCHK(II),(DUM2(J),J=2,8))
IF (I.EQ.NFWD) WRITE(6,276)
240 CONTINUE
DUM2(9) = WDUMP/CMSFL(IUNITS)
IF (ADUMP.GT.0.0.AND.IUNITS.EQ.2) WRITE(6,290) DUM2(9),WCHKDM
IF (ADUMP.GT.0.0.AND.IUNITS.EQ.1) WRITE(6,291) DUM2(9),WCHKDM

C
1000 CONTINUE
C
C TO DETERMINE THE MEAN OUTSIDE SURFACE TEMPERATURE FOR EACH
C SIDE OF THE BLADE, AND
C TO LOCATE THE EXTREME TEMPERATURE POINTS, BOTH HIGH AND LOW.
C
XTOT = 0.
XTOTMD = 0.0
HBAR = HC(1)*(DLX(2)+DLX(3))/2.
TBAR = T(2,ISLICE,1)*(DLX(6)+DLX(11))/2.
TBARM = T(2,ISLICE,3)*(DLX(8)+DLX(13))/2.
ISTAT = 1
DO 1004 I = 2,NSTA
NODM = 5*I-2
ISTAT = ISTAT + 5
ISTATD = ISTAT + 10
TBARM = TBARM + T(2,ISLICE,NODM)*(DLX(NODM)+DLX(NODM+10))/2.
XTOTMD = XTOTMD + DLX(NODM)
TBAR = TBAR + T(2,ISLICE,ISTAT)*(DLX(ISTAT)+DLX(ISTATD))/2.
XTOT = XTOT + DLX(ISTAT)
HBAR = HBAR + HC(I)*(DLX(ISTAT)+DLX(ISTATD))/2.
1004 CONTINUE
IF (IUNITS.EQ.1) TBAR = TBAR/(1.8*XTOT)
IF (IUNITS.EQ.2) TBAR = TBAR/XTOT - 460.
IF (IUNITS.EQ.1) TBARM = TBARM/(1.8*XTOTMD)
IF (IUNITS.EQ.2) TBARM = TBARM/XTOTMD - 460.
TBULK = TBULK + TBARM*S(ISLICE)
TOTSPN = TOTSPN + S(ISLICE)
HBAR = HBAR/(XTOT*CHTC(IUNITS))
1008 IF (IUNITS.EQ.2) WRITE(6,1115) TBAR,TBARM,HBAR
IF (IUNITS.EQ.1) WRITE(6,1116) TBAR,TBARM,HBAR
C
TSMAX = T(2,ISLICE,1) - 460.

```

```

NWROTTT 5641
NWROTTT 5642
NWROTTT 5643
NWROTTT 5644
NWROTTT 5645
NWROTTT 5646
NWROTTT 5647
NWROTTT 5648
NWROTTT 5649
NWROTTT 5650
NWROTTT 5651
NWROTTT 5652
NWROTTT 5653
NWROTTT 5654
NWROTTT 5655
NWROTTT 5656
NWROTTT 5657
NWROTTT 5658
NWROTTT 5659
NWROTTT 5660
NWROTTT 5661
NWROTTT 5662
NWROTTT 5663
NWROTTT 5664
NWROTTT 5665
NWROTTT 5666
NWROTTT 5667
NWROTTT 5668
NWROTTT 5669
NWROTTT 5670
NWROTTT 5671
NWROTTT 5672
NWROTTT 5673
NWROTTT 5674
NWROTTT 5675
NWROTTT 5676
NWROTTT 5677
NWROTTT 5678
NWROTTT 5679
NWROTTT 5680
NWROTTT 5681
NWROTTT 5682
NWROTTT 5683
NWROTTT 5684
NWROTTT 5685
NWROTTT 5686
NWROTTT 5687
NWROTTT 5688
NWROTTT 5689
NWROTTT 5690
NWROTTT 5691
NWROTTT 5692
NWROTTT 5693
NWROTTT 5694
NWROTTT 5695
NWROTTT 5696
NWROTTT 5697
NWROTTT 5698
NWROTTT 5699
NWROTTT 5700

```

TSMIN = T(2,ISLICE,1) - 460.	NWROTTT 5701
TPMAX = T(2,ISLICE,1) - 460.	NWROTTT 5702
TPMIN = T(2,ISLICE,1) - 460.	NWROTTT 5703
ISUCHX = 1	NWROTTT 5704
ISUCHN = 1	NWROTTT 5705
IPRSMX = 1	NWROTTT 5706
IPRSMN = 1	NWROTTT 5707
IPRES = 1	NWROTTT 5708
ISUCT = -4	NWROTTT 5709
C	NWROTTT 5710
DO 1080 I = 3,NSTA,2	NWROTTT 5711
IPRES = IPRES + 10	NWROTTT 5712
ISUCT = ISUCT + 10	NWROTTT 5713
C	NWROTTT 5714
C	NWROTTT 5715
IF (T(2,ISLICE,IPRES)-460..GT.TPMAX) GO TO 1030	NWROTTT 5716
IF (T(2,ISLICE,IPRES)-460..LT.TPMIN) GO TO 1040	NWROTTT 5717
GO TO 1050	NWROTTT 5718
C	NWROTTT 5719
1030 TPMAX = T(2,ISLICE,IPRES) -460.	NWROTTT 5720
IPRSMX = I	NWROTTT 5721
GO TO 1050	NWROTTT 5722
C	NWROTTT 5723
1040 TPMIN = T(2,ISLICE,IPRES) -460.	NWROTTT 5724
IPRSMN = I	NWROTTT 5725
C	NWROTTT 5726
1050 IF (T(2,ISLICE,ISUCT)-460..GT.TSMAX) GO TO 1060	NWROTTT 5727
IF (T(2,ISLICE,ISUCT)-460..LT.TSMIN) GO TO 1070	NWROTTT 5728
GO TO 1080	NWROTTT 5729
C	NWROTTT 5730
1060 TSMAX = T(2,ISLICE,ISUCT) -460.	NWROTTT 5731
ISUCHX = I - 1	NWROTTT 5732
GO TO 1080	NWROTTT 5733
C	NWROTTT 5734
1070 TSMIN = T(2,ISLICE,ISUCT) -460.	NWROTTT 5735
ISUCHN = I - 1	NWROTTT 5736
1080 CONTINUE	NWROTTT 5737
C	NWROTTT 5738
C	NWROTTT 5739
IF (ISLICE.LT.NSLICE) GO TO 1095	NWROTTT 5740
TBULK = TBULK/TOTSPN	NWROTTT 5741
IF (IUNITS.EQ.1) WRITE(6,1091) TYME,TBULK	NWROTTT 5742
IF (IUNITS.EQ.2) WRITE(6,1090) TYME,TBULK	NWROTTT 5743
C	NWROTTT 5744
1090 FORMAT(1H2,30X,'TIME =',F6.3,' SEC., OVERALL BULK TEMPERATURE =',	NWROTTT 5745
Z F7.1,' F')	NWROTTT 5746
1091 FORMAT(1H2,30X,'TIME =',F6.3,' SEC., OVERALL BULK TEMPERATURE =',	NWROTTT 5747
Z F7.1,' K')	NWROTTT 5748
1095 CONTINUE	NWROTTT 5749
1115 FORMAT(/' OVERALL AREA WEIGHTED AVERAGES--OUTSIDE T =',F7.1,	NWROTTT 5750
Z ' F, MID-WALL T =',F7.1,	NWROTTT 5751
Z ' F, COOLANT H = 'F6.1,' BTU/(HR/FT**2/R)')	NWROTTT 5752
1116 FORMAT(/' OVERALL AREA WEIGHTED AVERAGES--OUTSIDE T =',F7.1,	NWROTTT 5753
Z ' K, MID-WALL T =',F7.1,	NWROTTT 5754
Z ' K, COOLANT H = 'F6.1,' WATTS/M**2/K')	NWROTTT 5755
IF (IUNITS.EQ.2) WRITE(6,1120) TPMAX,IPRSMX,TPMIN,IPRSMN,TSMAX,	NWROTTT 5756
Z ISUCHX,TSMIN,ISUCHN	NWROTTT 5757
1120 FORMAT(/12X,'EXTREMES CP OUTER SURFACE TEMPERATURES (F)'/6X,	NWROTTT 5758
Z 'PRESSURE SIDE: ',	NWROTTT 5759
Z F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2/6X,	NWROTTT 5760

```

Z      'SUCTION SIDE: ',
Z      F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2)
IF (IUNITS.EQ.2) GO TO 1130
TPMAX = (TPMAX+460.)/1.8
TPMIN = (TPMIN+460.)/1.8
TSMAX = (TSMAX+460.)/1.8
TSMIN = (TSMIN+460.)/1.8
WRITE(6,1125) TPMAX,IPRSMX,TPMIN,IPRSMN,TSMAX,ISUCHX,TSMIN,ISUCHN
1125  FORMAT(/12X,'EXTREMES OF OUTER SURFACE TEMPERATURES (K)'/6X,
Z      'PRESSURE SIDE: ',
Z      F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2/6X,
Z      'SUCTION SIDE: ',
Z      F7.1,' AT STATION ',I2,', ',F7.1,' AT STATION ',I2)
1130  CONTINUE
100   FORMAT(30X,'PRESSURE SIDE , TRAILING EDGE REGION BEGINS AT ',
Z      'STATION-',I3)
124   FORMAT(1H1,'/' SLICE NO. ',I2,17X,'SUCTION SIDE , TRAILING EDGE ',
Z      'REGION BEGINS AT STATION-',I3)
8806  FORMAT(' SLICE NO.',I2,', FLOW SPLIT NO.',I3,', SPLIT AT ',
Z      'STATION',I3,
Z      '; FRACTION SPLIT TO SUCTION SIDE IS',F7.4,I6,' ITERATIONS')
270   FORMAT(/
Z      ' STATION*COOLANT* OUTSIDE *INTERFACE* MID-WALL* INSIDE * ',
Z      'COOLANT * STATIC P* TOTAL P * HC,BTU/HR* HG,BTU/HR* TG'
Z/' NUMBER *NODE NO* T (F) * T (F) * T (F) * T (F) * T ',
Z      '(F) * (PSIA) * (PSIA) * /FT**2/R * /FT**2/R * (F) '
Z/119('**')/)
271   FORMAT(/
Z      ' STATION*COOLANT* OUTSIDE *INTERFACE* MID-WALL* INSIDE * ',
Z      'COOLANT * STATIC P* TOTAL P * HC * HG * TG'
Z/' NUMBER *NODE NO* T (K) * T (K) * T (K) * T (K) * T ',
Z      '(K) * (KPA) * (KPA) * W/M**2/K * W/M**2/K * (K) '
Z/119('**')/)
274   FORMAT(I6,2X,I6,1X,7F10.1,3F11.1)
276   FORMAT(47X,'BEGIN TRAILING EDGE REGION')
278   FORMAT(1H2,'/' STATION * COOLANT * IMP. FLOW * RE-NO. * CROSSFLOW',
Z      ' * RE-NO. * MACH NO., '
Z      ' FRICTION * FILM FLOW * EFFECTIVENESS **/'
Z      ' NUMBER * NODE NO * (LBM/SEC) * JET * (LBM/SEC) ',
Z      '** CROSSFLOW * CROSSFLOW **'
Z      ' FACTOR * (LBM/SEC) **/115('**')/19X,'**',20X,'**',46X,'**')
279   FORMAT(1H2,'/' STATION * COOLANT * IMP. FLOW * RE-NO. * CROSSFLOW',
Z      ' * RE-NO. * MACH NO., '
Z      ' FRICTION * FILM FLOW * EFFECTIVENESS **/'
Z      ' NUMBER * NCDE NO * (KG/SEC) * JET * (KG/SEC) ',
Z      '** CROSSFLOW * CROSSFLOW **'
Z      ' FACTOR * (KG/SEC) **/115('**')/19X,'**',20X,'**',46X,'**')
280   FORMAT(1H ,I5,5X,I5,' ',F9.6,A1,P9.1,' * ',F9.6,2X,P9.1,4X,P9.6
Z      ,2X,F9.6,' * ',F9.6,4X,F9.6)
290   FORMAT(/15X,' FLOW DUMPED DIRECTLY INTO TRAILING EDGE REGION IS '
Z      ,F10.6,A1,' LBM/SEC')
291   FORMAT(/15X,' FLOW DUMPED DIRECTLY INTO TRAILING EDGE REGION IS '
Z      ,F10.6,A1,' KG/SEC')
      RETURN
      END

```

REFERENCES

1. Gaugler, Raymond E.: TACT1, A Computer Program for the Transient Thermal Analysis of a Cooled Turbine Blade or Vane Equipped with a Coolant Insert. I - Users Manual. NASA TP-1271, 1978.
2. Poferl, David J.; and Svehla, Roger A.: Thermodynamic and Transport Properties of Air and Its Products of Combustion with ASTM-A-1 Fuel and Natural Gas at 20, 30, and 40 Atmospheres. NASA TN D-7488, 1973.
3. Stollery, J. L.; and El-Ehwany, A. A. M.: A Note on the Use of a Boundary-Layer Model for Correlating Film-Cooling Data. Int. J. Heat Mass Transfer, vol. 8, 1965, pp. 55-65.

TABLE I. - SUBROUTINE CALLS AND COMMON BLOCKS

Subroutine name	Source module	COMMON blocks	Called subroutines	Calling subroutines	Subroutine name	Source module	COMMON blocks	Called subroutines	Calling subroutines
BLOCK DATA	NGASDAT	/GAAS/	NONE	NONE	PARRAY	NPARAYT	/MATRX/ /PRPS/ /TCO/ /TRNSNT/	NONE	TCOEF
FLWS	NFLOEST	/CHKHOL/ /FLMCOL/ /FRIC/ /PRPS/ /TCO/ /TRNSNT/	GASTBL	TCOEF	PLNUM	NPLENMP	/RADL/ /TCO/ /TRNSNT/ /UNITS/	GASTBL	MAIN PROGRAM
FLSPLT	NFLSPLP	NONE	NONE	TCOEF	PLOTMF	NPLOTIT	/PRPS/ /SPECL/ /TCO/ /TRNSNT/ /UNITS/	NONE	TCOEF MAIN PROGRAM
GASTBL	NGASTB	/GAAS/	NONE	FLWS HCFRCD HCOOL HCPINS PLNUM TARRAY	PREP	NPREPAT	/BOUND/ /FLMCOL/ /FRIC/ /PRPS/ /SPECL/ /TCO/ /TRNSNT	NONE	INPRT MAIN PROGRAM
GAUSS	NGAUS	/MATRX/	NONE	TCOEF	TARRAY	NTARAYT	/BOUND/ /FLMCOL/ /MATRX/ /PRPS/ /TCO/ /TRNSNT/	GASTBL HCFRCD HCPINS	TCOEF
GETIN	NGETINT	/BOUND/ /FLMCOL/ /IMPCOR/ /RADL/ /SPECL/ /TCO/ /TRNSNT/ /UNITS	INPRT	MAIN PROGRAM	TCOEF	NTCOFTT	/MATRX/ /PRPS/ /TCO/ /TRNSNT/	FLWS FLSPLT GAUSS HCOOL PARRAY PLOTMF TARRAY THRCON WROUT	MAIN PROGRAM
HCFRCD	NHCFRCT	/TCO/	GASTBL	HCOOL TARRAY	THRCON	NTHRCNT	/BOUND/ /TCO/	NONE	TCOEF
HCOOL	NHCOOLT	/IMPCOR/ /PRPS/ /TCO/	GASTBL HCFRCD	TCOEF	WROUT	NWROTTT	/CHKHOL/ /FLMCOL/ /PRPS/ /RADL/ /TCO/ /TRNSNT/ /UNITS/	NONE	TCOEF
HCPINS	NHCPINT	/PRPS/ /TCO/	GASTBL	TARRAY					
INPRT	NINPRTT	/BOUND/ /GAAS/ /PRPS/ /RADL/ /SPECL/ /TCO/ /TRNSNT/ /UNITS/	PREP	GETIN					
MAIN PROGRAM	NTTACT	/BOUND/ /FLMCOL/ /GAAS/ /RADL/ /SPECL/ /TCO/ /TRNSNT/ /UNITS/	GETIN PLNUM PLOTMF PREP TCOEF	NONE					

TABLE II. - COMMON-BLOCK CROSS-REFERENCE TABLE

Subroutine	COMMON block												
	BOUND	CHKHOL	FLMCOL	FRIC	GAAS	IMPCOR	MATRX	PRPS	RADL	SPECL	TCO	TRNSNT	UNITS
BLOCK DATA					×								
FLWS		×	×	×				×			×	×	
FLSPLT													
GASTBL					×								
GAUSS							×						
GETIN	×		×			×			×	×	×	×	×
HCFRCD											×		
HCOOL						×		×			×		
HCPINS								×			×		
INPRT	×				×			×	×	×	×	×	×
NTTACT	×		×		×				×	×	×	×	×
PARRAY							×	×			×	×	
PLNUM									×		×	×	×
PLOTMF								×		×	×	×	×
PREP	×		×	×				×		×	×	×	
TARRAY	×		×				×	×			×	×	
TCOEF							×	×			×	×	
THRCON	×										×		
WROUT		×	×					×	×		×	×	×

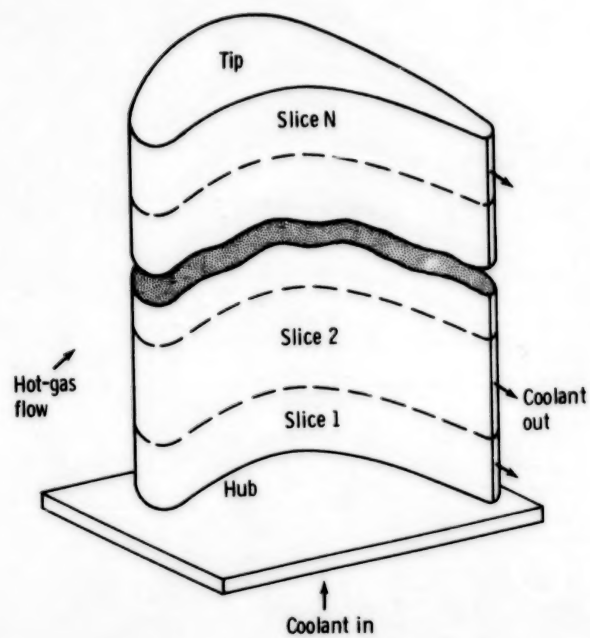


Figure 1. - Overall division of blade into slices.

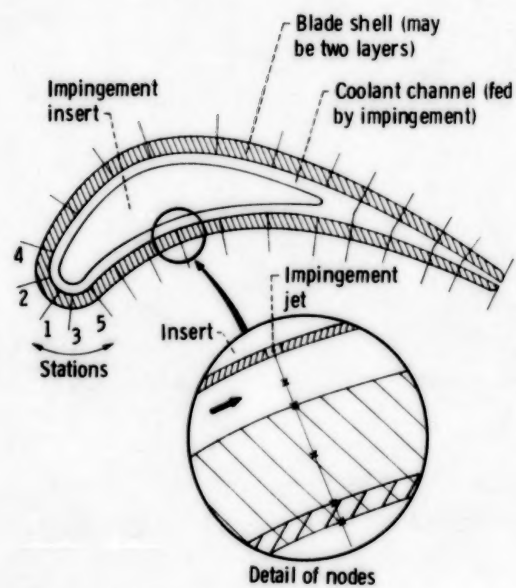


Figure 2 - Blade geometric model.

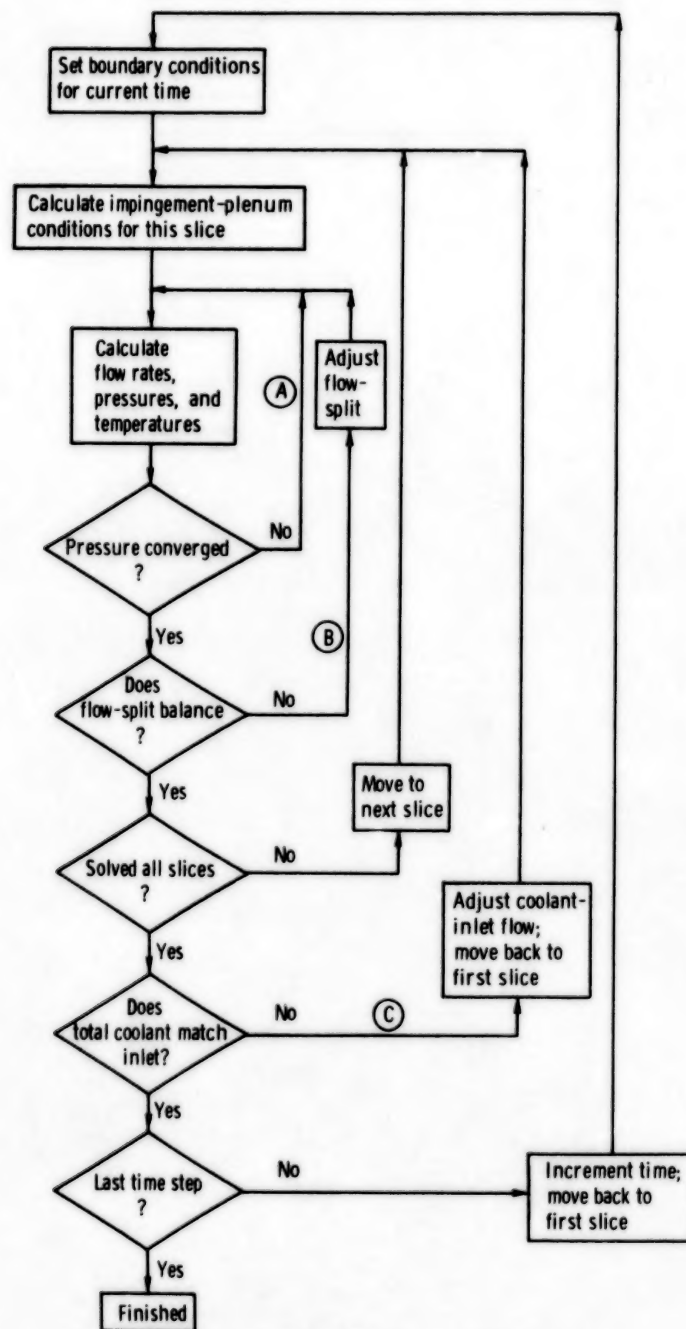


Figure 3. - Overall program procedure.

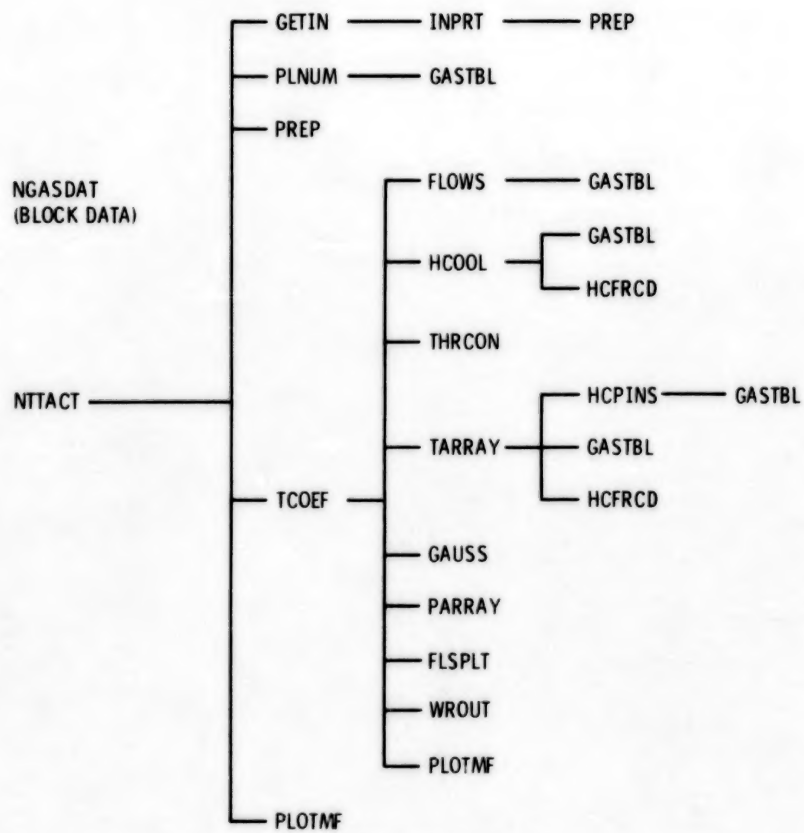


Figure 4 - Subroutine calling relations.

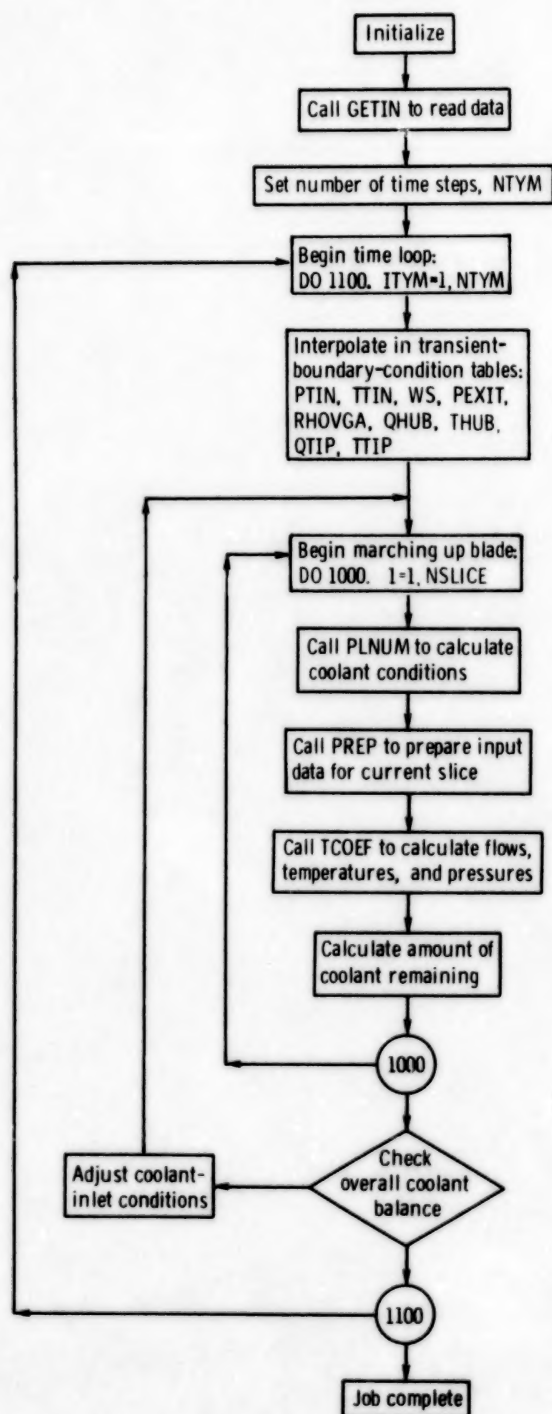


Figure 5. - Flow chart of main program.

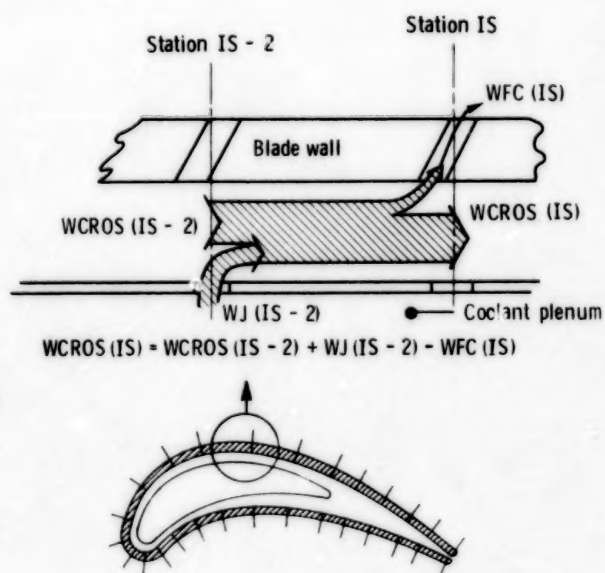


Figure 6. - Coolant-channel mass balance.

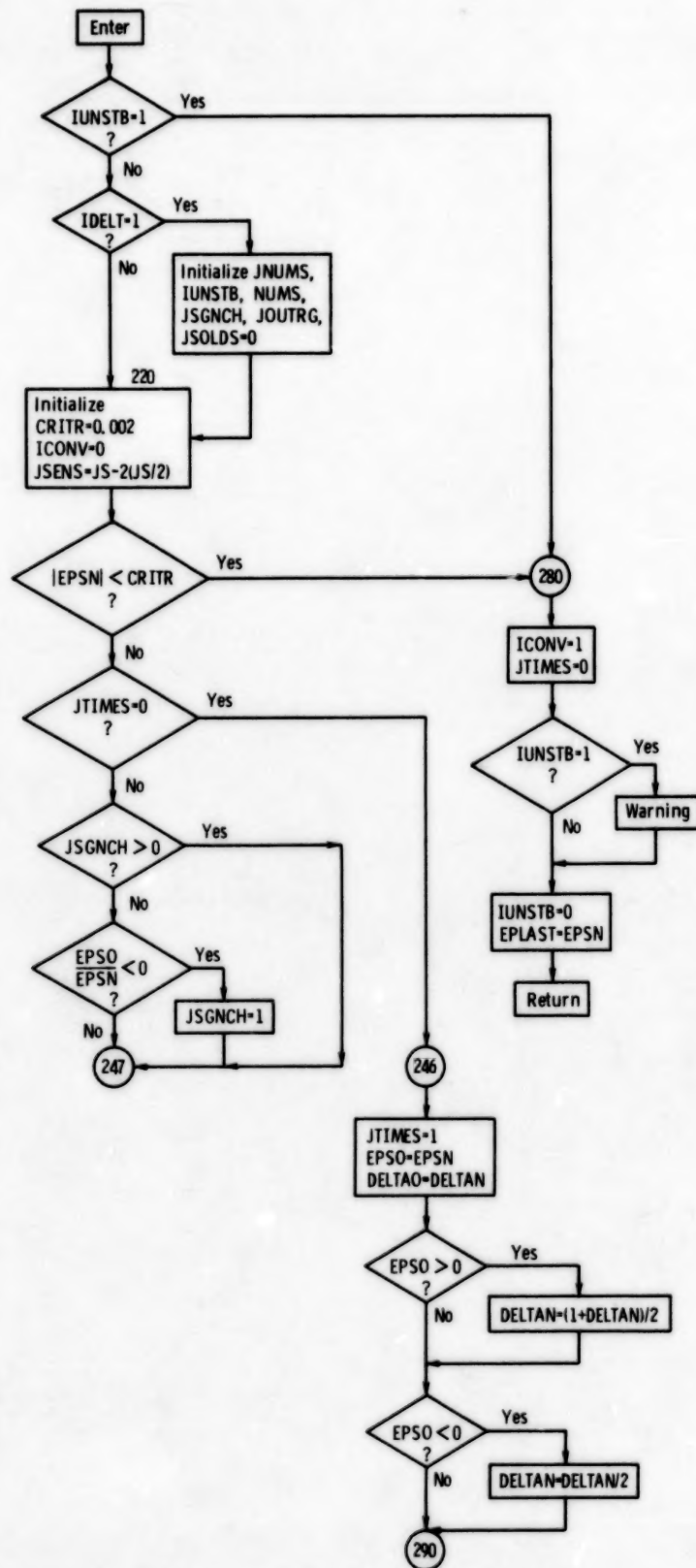


Figure 7. - Flow chart for subroutine FLSPLT.

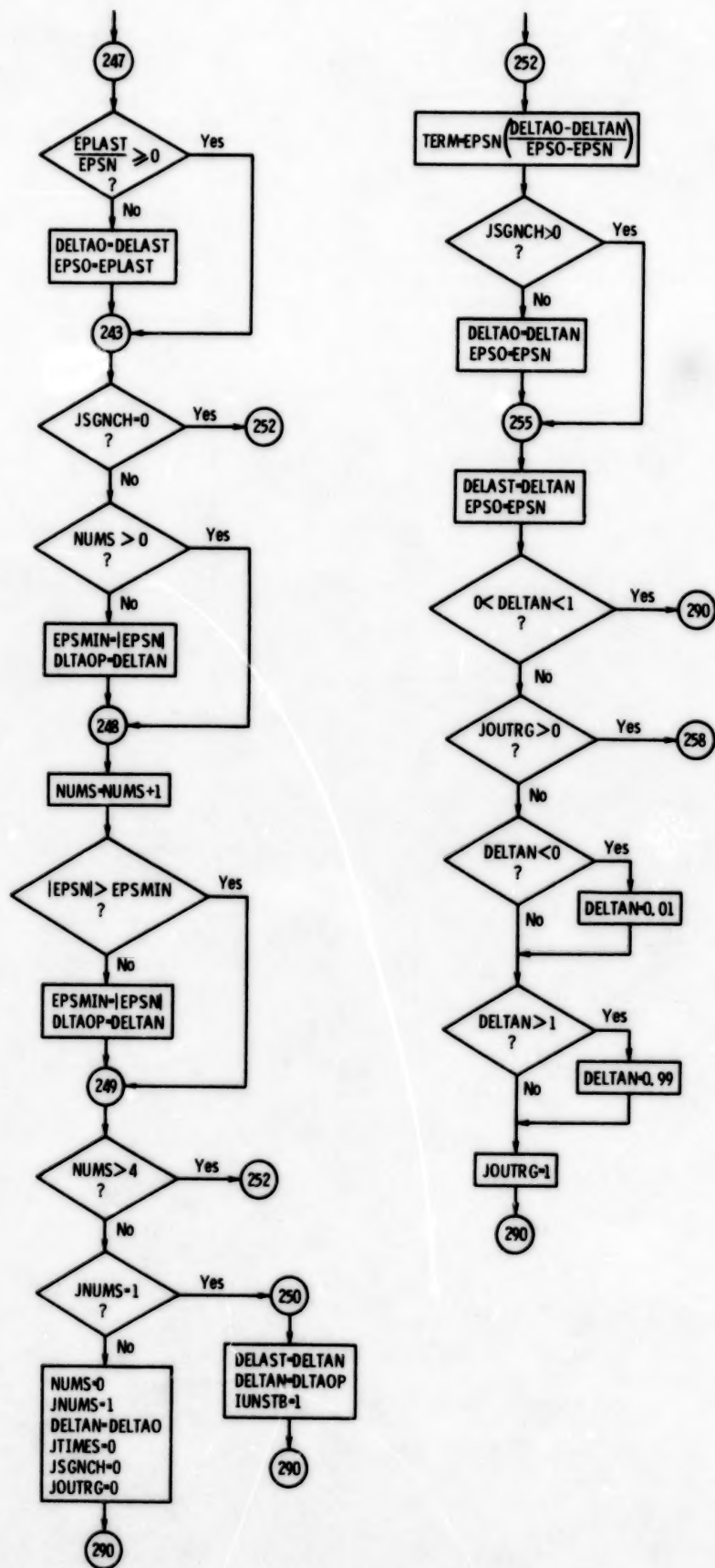


Figure 7. - Continued.

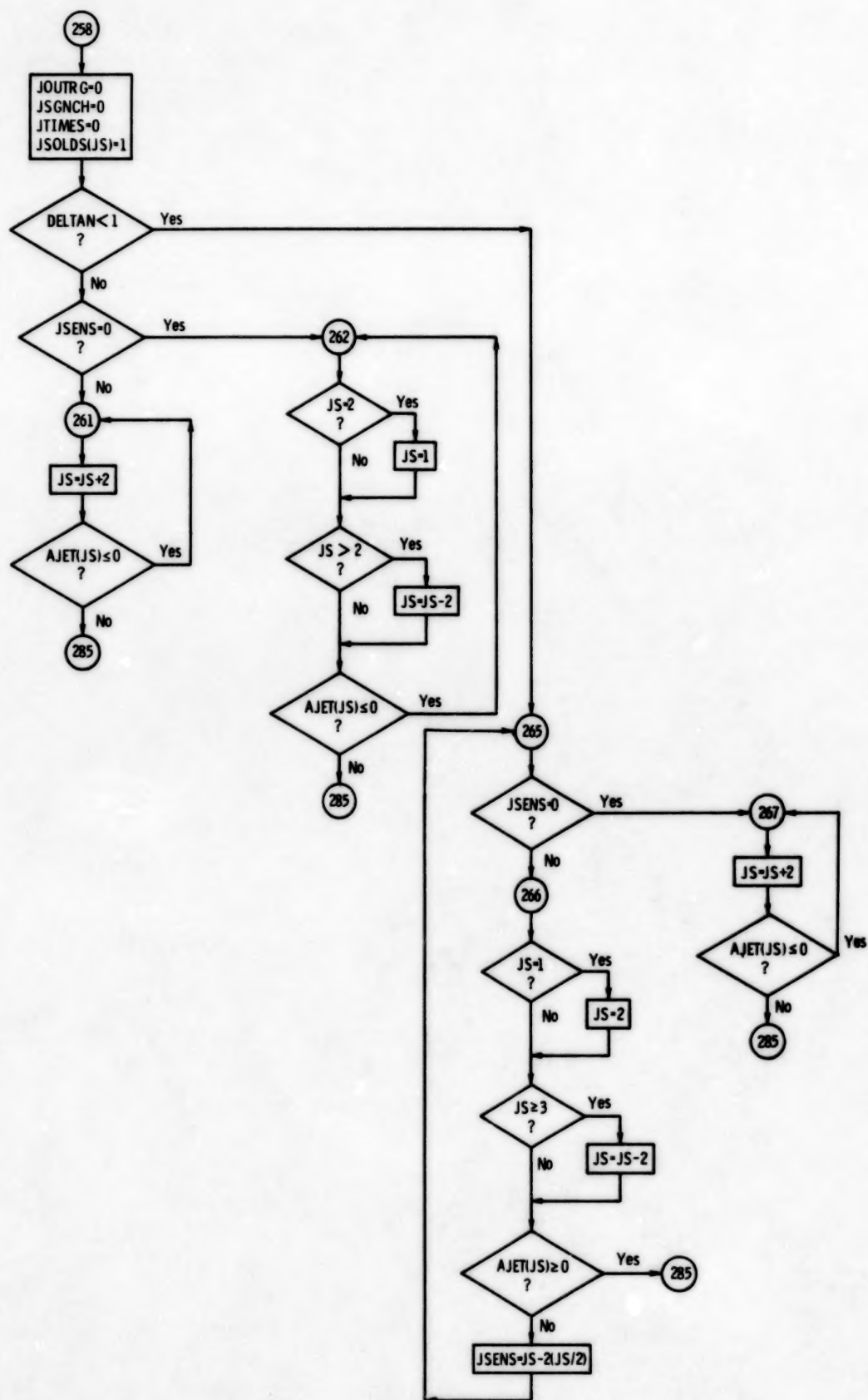


Figure 7. - Continued.

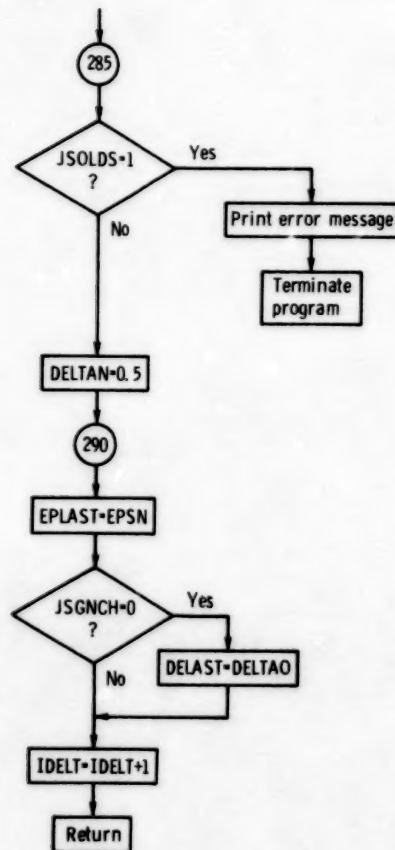
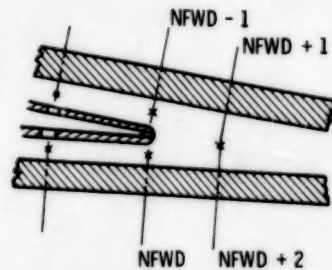
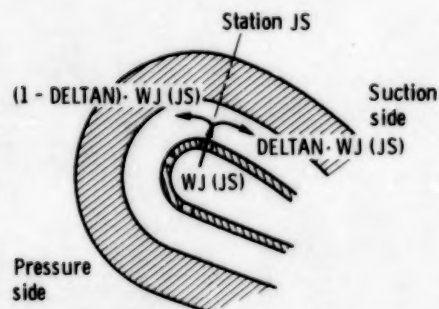


Figure 7. - Concluded.



(a) Detail of station numbering at end of impingement insert.



(b) Detail of flow-split point.

Figure 8. - Details of flow-split parameters.

1. Report No. NASA TP-1391	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle TACT1, A COMPUTER PROGRAM FOR THE TRANSIENT THERMAL ANALYSIS OF A COOLED TURBINE BLADE OR VANE EQUIPPED WITH A COOLANT INSERT. II - PROGRAMMERS MANUAL		5. Report Date January 1979	
		6. Performing Organization Code	
7. Author(s) Raymond E. Gaugler		8. Performing Organization Report No. E-9767	
		10. Work Unit No. 505-04	
9. Performing Organization Name and Address National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135		11. Contract or Grant No.	
		13. Type of Report and Period Covered Technical Paper	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D. C. 20546		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract <p>A computer program to calculate transient and steady-state temperatures, pressures, and coolant flows in a cooled axial-flow turbine blade or vane with an impingement insert is described. Coolant-side heat-transfer coefficients are calculated internally in the program, with the user specifying either impingement or convection heat transfer at each internal flow station. Spent impingement air flows in a chordwise direction and is discharged through the trailing edge and through film-cooling holes. The ability of the program to handle film cooling is limited by the internal flow model. Input to the program includes a description of the blade geometry, coolant-supply conditions, outside thermal boundary conditions, and wheel speed. The blade wall can have two layers of different materials, such as a ceramic thermal-barrier coating over a metallic substrate. Program output includes the temperature at each node, the coolant pressures and flow rates, and the coolant-side heat-transfer coefficients.</p>			
17. Key Words (Suggested by Author(s)) Heat transfer Turbine cooling Computer program Impingement cooling		18. Distribution Statement Unclassified - unlimited STAR Category 34	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 163	22. Price* A08

END

September 17, 1979